Forschungsrahmenprogramm der Bundesregierung für IT-Sicherheit – Selbstbestimmt und sicher in der digitalen Welt

Verbundprojekt:

Automatisierte Bedrohungsmodellierung für den sicheren Softwarelebenszyklus - OVVL

Teilvorhaben: ***OVVL - Komponentenbasiert, Offen und Modular - KOM***

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Abschlussbericht Hochschule Offenburg

Förderkennzeichen: **16KIS1403**
Laufzeit des Vorhabens:
01.07.2021 bis 30.06.2024

Zuwendungsgeber:
Bundesministerium für
Bildung und Forschung

Zuwendungsempfänger:
Hochschule Offenburg

Hochschule Offenburg
offenburg.university

Prof. Dr. Andreas Schaad Offenburg, 19.12.2024

Inhalt

1 Zusammenfassung 1

2 Aufzählung der wichtigsten wissenschaftlich-technischen Ergebnisse und anderer wesentlicher Ereignisse 5

2.1 AP 1: Erhebung der Fallstudien und Anforderungen (4PM) 5

2.2 AP 2: Entwurf OVVL Methodik und Datenmodell (8PM) 7

2.3 AP 3: Entwurf OVVL Komponentenmodell (16 PM) 12

2.4 AP 4: Verwendung von OVVL in Sicherheitsprozessen (0 PM) 15

2.5 AP 5: Entwurf Prozess- und Betriebsmodell (0 PM) 15

2.6 AP 6: Entwurf OVVL Plattform & Integration (6 PM) 15

2.7 AP 7: Validierung, Test und Evaluation (6 PM) 17

3 Vergleich des Vorhabensstands mit der ursprünglichen (bzw. mit Zustimmung des Zuwendungsgebers geänderten) Arbeits-, Zeit- und Ausgabenplanung 18

4 Fortschreibung des Verwertungsplans 18

4.1 Wissenschaftliche und/oder technische Erfolgsaussichten nach Projektende (mit Zeithorizont) 18

4.2 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit für eine mögliche notwendige nächste Phase bzw. die nächsten innovatorischen Schritte zur erfolgreichen Umsetzung der Ergebnisse. 18

5 Anhang: Screenshots OVVL Proof of Concept 18

1 Zusammenfassung

Das Projekt „OVVL“ wurde erfolgreich beendet. Ziel des Projekts OVVL war die automatisierte Unterstützung eines „Secure by Design“ Ansatzes, der den gesamten sicheren Softwarelebenszyklus adressiert. Dafür wurde folgende drei Teilprobleme gelöst:

1. Erstellung einer Methodik und unterstützender Tools für automatisierte (machine-learning basierte) Bedrohungsanalysen von Softwarearchitekturen in der Designphase.
2. Entwurf von Datenstrukturen und offenen Schnittstellen, um die Ergebnisse solch einer Analyse auch von automatisierten Verfahren in späteren Phasen des sichereren Softwarelebenszyklus nutzen zu können (insb. Testen und Vulnerability Management).
3. Entwurf eines Prozessportfolios, um Methodik, Tooling und Schnittstellen in domänenspezifische Bedrohungsanalysen (Automatisierungstechnik, Gesundheitswesen, IT, …) und den sicheren Entwurf genereller Software einbinden zu können.

Diese Arbeiten resultieren in dem OVVL Datenmodell und Tool, welches nun durch die beteiligten Partner ENRW und Secuvera validiert und ggf. verstetigt wird:

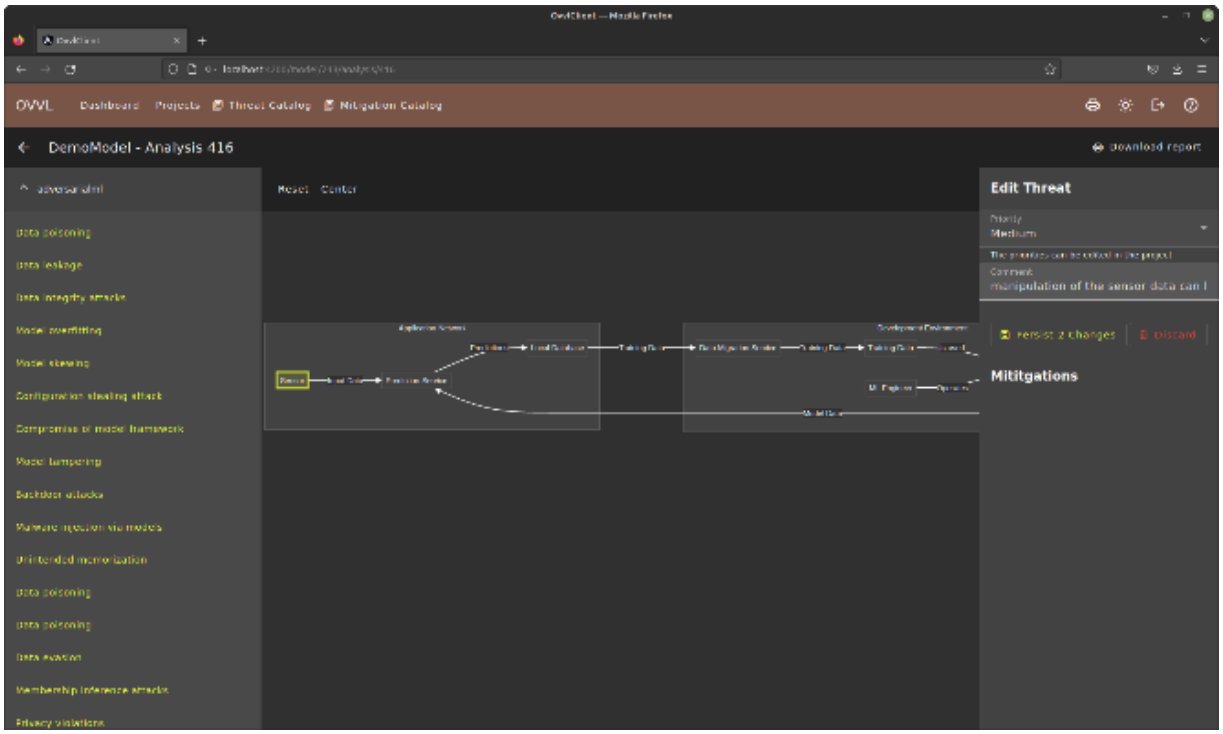


Abbildung: OVVL Tool

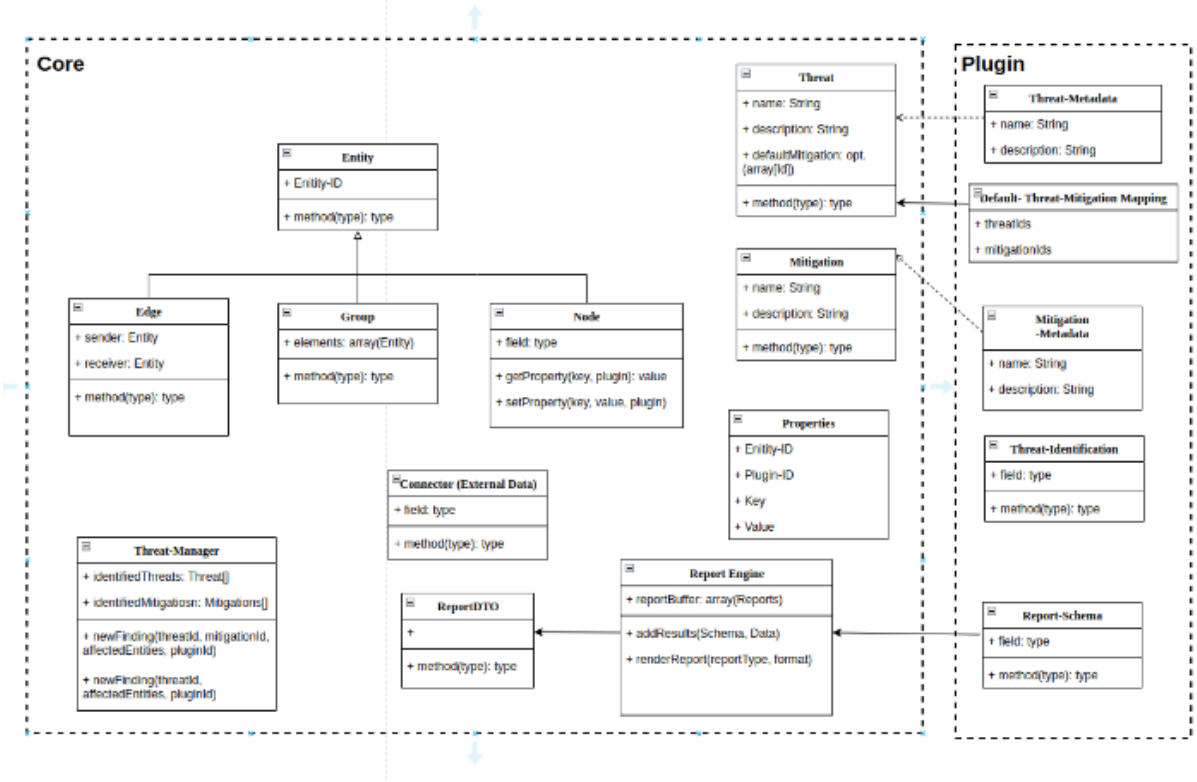


Abbildung: OVVL Datenmodell

Folgende Veröffentlichungen wurden zudem aus dem Projekt generiert:

1. Andreas Schaad, Dominik Binder: Deep-Learning-based Vulnerability Detection in Binary Executables. 15th International Symposium on Foundations & Practice of Security (FPS 2022), Ottawa, Kanada
2. Andreas Schaad, Dominik Binder: Deep-Learning-based Vulnerability Detection in Binary Executables. CoRR abs/2212.01254 (2022) – erweiterte Version

Ein in Teilen (KI-basierte Threat Analyse) auf OVVL basierendes Folgeprojekt wurde durch die HSO erfolgreich initiiert (3 jähriges KMU Innovativ Projekt „SMILE“ FKZ 16KIS1929, 01.08.2023-31.07.2026).

Die Verteilung der Projekt-Arbeitsaufwände auf die einzelnen Arbeitspakete, die HSO (hervorgehoben) und die Verbundpartner ist der folgenden Tabelle zu entnehmen (Angabe Personenmonaten):

AP	Inhalt	*HSO*	ERNW	SECU	Summe

AP 1	Erhebung der Fallstudien und Anforderungen				
AP 1.1	Aufstellen Nutzergruppe: Definition domänen-spezifischer Fallstudien sowie Anforderungen / Referenzsysteme für technische Validierung	*1*	6	3	10
AP 1.2	Übersicht & Evaluierung von aktuellen Threat Modelling Ansätzen	*3*	3	2	8
AP 1.3	Definition der Validierungsparameter für späteren Einsatz in produktiven Umgebungen	*0*	3	4	7
AP 2	Entwurf OVVL Methodik und Datenmodell				
AP 2.1	Refactoring existierende Methodik, Tooling und open-source Entwick-lungsprozess	*3*	2	0	5
AP 2.2	Design und Implementierung OVVL Datenmodell und offene APIs	*3*	4	0	7
AP 3	Entwurf OVVL Komponentenmodell				
AP 3.1	Design OVVL Komponentenmodell	*2*	2	1	5
AP 3.2	Modul 1: Attack Trees / Misuse Cases	*2*	6	2	10
AP 3.3	Modul 2: Identifikation von Datenflüssen	*0*	6	0	6
AP 3.4	Modul 3: KI-gestützte Threat Analyse	*5*	2	2	9
AP 3.5	Modul 4: DSGVO / GDPR Analyse	*5*	1	0	6
AP 4	Verwendung von OVVL in Sicherheitsprozessen				
AP 4.1	Betrachtung von Fallstudien bei der Nutzung von OVVL in SDL	*0*	3	6	9
AP 4.2	Entwicklung von integrierten Threat Modelling Prozessen	*0*	3	6	9
AP 5	Entwurf OVVL Prozess und Betriebsmodell				
AP 5.1	Betriebswirtschaftliche Evaluierung möglicher Fallstudien	*0*	4	3	7
AP 5.2	Simulation und Zwischenevaluation	*0*	2	3	5
AP 6	Integration OVVL Plattform & Demonstrator				
AP 6.1	Bereitstellung als Clouddienst (plattformunabhängig)	*3*	3	0	6
AP 6.2	Einbindung individueller Module	*3*	3	0	6
AP 7	Validierung, Test und Evaluation				
AP 7.1	Validierung von Annahmen und Thesen aus dem Vorhaben	*3*	4	4	11
AP 7.2	Domänenspezifische Evaluation	*3*	7	6	16
Gesamt		*36*	64	42	142

Abbildung: Arbeitspakete und Aufwände (HSO hervorgehoben vgl. GANTT Planung)

Die Abhängigkeiten zwischen den Arbeitspaketen stellen sich wie in Abbildung 2 dar (Die Arbeitspakte der HSO sind grau hinterlegt). Wesentlich sind die in Abbildung 3 zu dem jetzigen Zeitpunkt geplanten personellen Aufwendungen, und die folgenden definierten Gesamtmeilensteine sowie damit prüfbare Abbruchkriterien des Projekts:

- M12: Exemplarische Fallstudien / Anforderungen und messbare technische und betriebswirtschaftliche KPIs für spätere Validierung und Verifikation
- M24: Erster Demonstrator der insbesondere den OpenAPI Ansatz und modulare Systemarchitektur veranschaulicht
- M34: Zweiter finaler Demonstrator, Validierung und Verifikation

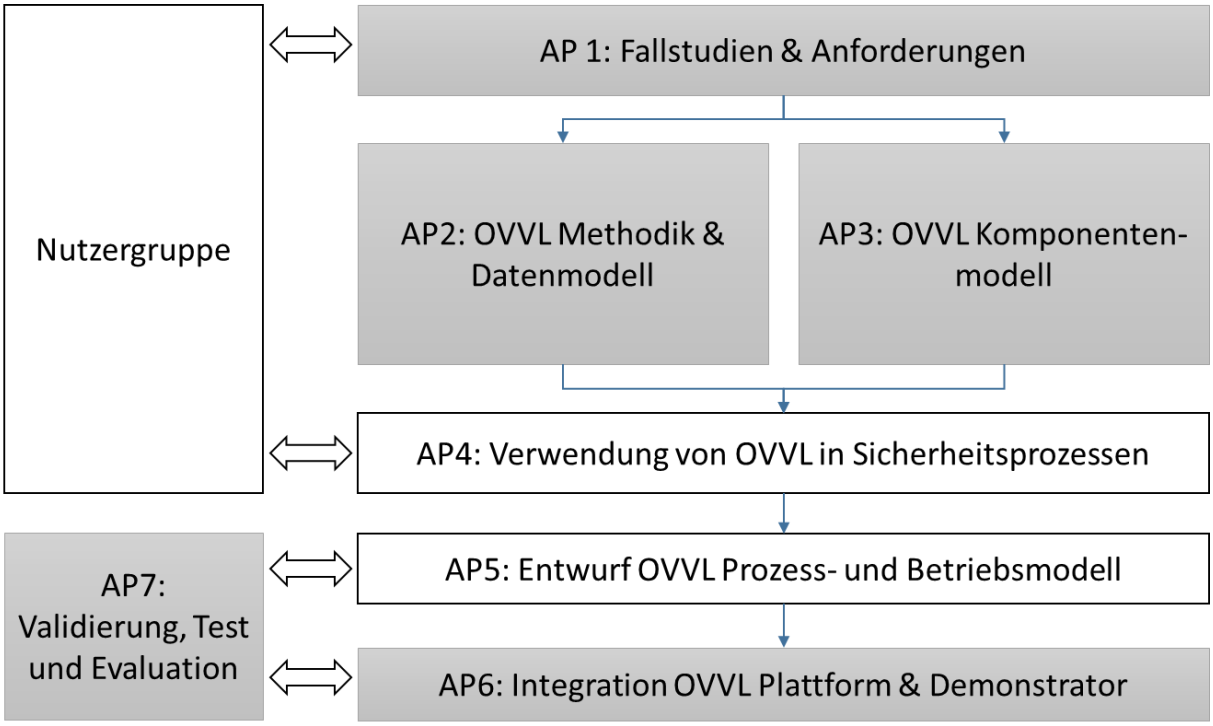


Abbildung: Struktur und Abhängigkeit der Arbeitspakete der HSO (grau hervorgehoben)

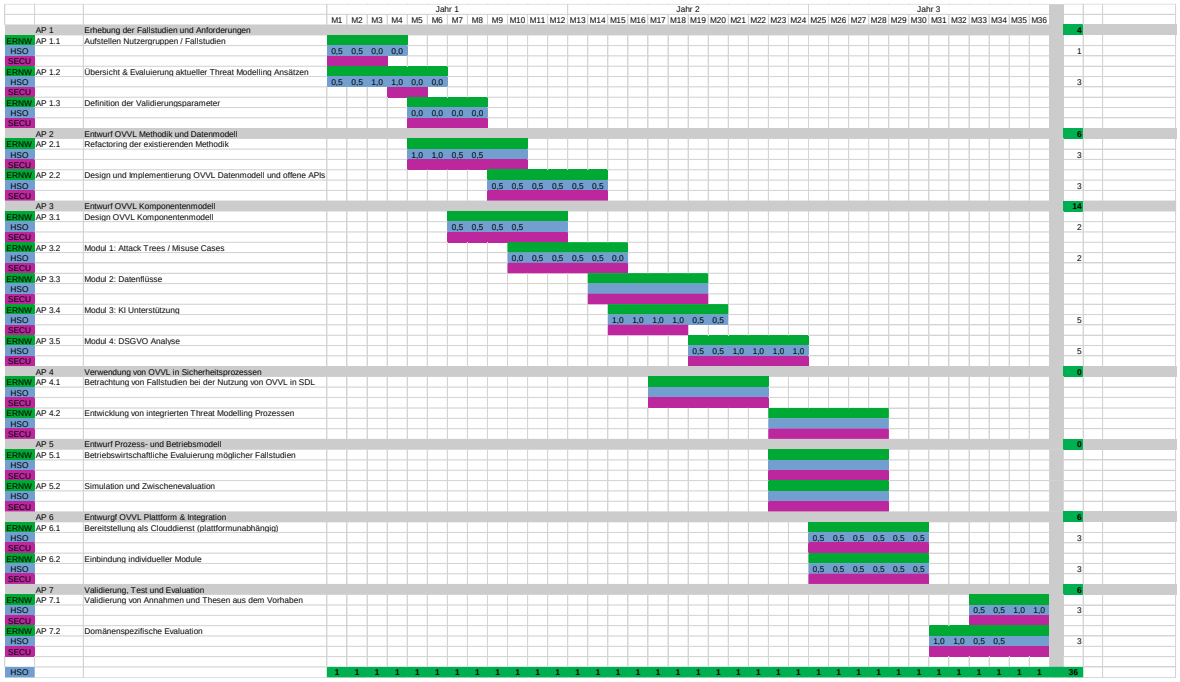


Abbildung: Projektplan für HSO mit detaillierter Ressourcenplanung

2 Aufzählung der wichtigsten wissenschaftlich-technischen Ergebnisse und anderer wesentlicher Ereignisse

2.1 AP 1: Erhebung der Fallstudien und Anforderungen (4PM)

Arbeiten und Ergebnisse aus AP1.1:

kein geplanter Aufwand für die HSO

Arbeiten und Ergebnisse aus AP 1.2:

Die HSO brachte hier in aufgearbeiteter Form die Daten aus dem in der Vergangenheit erfolgten Betrieb der bereits existierenden technischen OVVL Machbarkeitsstudie ein.

Die Arbeiten (AP1.2) an der Übersicht & Evaluierung von aktuellen Threat Modelling Ansätzen wurden durch die HSO in Form eines technischen Berichts erstellt. Inhalte wurde in Ansätzen auch in die Lehre an der HSO übernommen (Studiengang ENITS (MSc)).

Es wurde eine Übersicht & Evaluierung von aktuellen Threat Modelling Ansätzen erarbeitet, um so den wissenschaftlichen / technischen State of the Art zu erfassen:

Name	Focus	Automated identification	Risk assessment	Countermeasures	Type of modeling
STRIDE	Security Threats	✓		✓	DFD
VAST	Security Threats	✓	✓	✓	DFD + PFD
TRIKE	Security Threats	✓	✓	✓	DFD + Use Flow
DREAD	Risk assessment		✓		-
CVSS	Risk assessment		✓		-
PASTA	General framework		✓*	✓*	DFD
OCTAVE	General framework		✓*	✓*	-
LINDDUN	Privacy Threats	✓		✓	DFD + Attack Trees

* Not automated
DFD = Data-flow diagram
PFD = Process-flow diagram

Tabelle 1: Overview of threat modeling methods

Tool Name	Open Source	Commercial	ML Integrated	Extendable	Platform	Analysis methods
Microsoft Threat Modeling Tool					Desktop App. (Windows only)	STRIDE
Cairis	✓			✓	Webapp	?
IriusRisk		✓*			Webapp	?
ThreatModeler		✓*			Webapp	VAST
Kenna.VM		✓	✓		Webapp	?
OWASP Threat Dragon	✓				Webapp	STRIDE, LINDDUN, CIA
SD Elements		✓			Webapp	?
SecuriCAD		✓			Desktop App.	STRIDE, MITRE ATT&CK
Threagile (Beta Version)	✓			✓	IDE integrated	STRIDE + custom
Tutamen		✓			Webapp	STRIDE, OWASP Top 10, CWE, CAPEC
Threatspec	✓				Code annotation	?
OVVL	✓				Webapp	STRIDE, CVE, CWE

* Free community edition
? = unknown or custom analysis method

Tabelle 5: Threat Modeling Tools and Features

Arbeiten und Ergebnisse aus AP 1.3

kein geplanter Aufwand für die HSO

2.2 AP 2: Entwurf OVVL Methodik und Datenmodell (8PM)

Arbeiten und Ergebnisse aus AP 2.1:

Im Rahmen des AP2.1 wurde das bestehende OVVL Datenmodell mit den Anforderungen des AP1 abgeglichen. Aufgrund der ermittelten Defizite des existierenden Modells, wurde durch die Projektbeteiligten beschlossen, dass eine Neuentwicklung der Anwendung angemessen ist. Des Weiteren wurden gemeinsam mit den Projektpartnern die technischen Aspekte des Entwicklungsprozesses (Tooling, Technologie-Stack etc.) festgelegt.

Somit ergaben ich zusammenfassend im Rahmen des Anforderungsabgleichs der früheren Implementierung mit Anforderungen aus AP 1:

- Mangelnde Abstraktion der DFD-Elemente
- Fehlen einer zentralen Threat-Datenbank im Datenmodell
- Fehlende Abstraktion eines Threats (Mangelnde Erweiterbarkeit, Wartbarkeit)
- Erweiterbarkeit der Elemente, Attribute und Threats nicht gegeben
- Möglichkeit zum Festlegen des Sicherheitsniveaus und weitere Informationen der Elemente

Arbeiten und Ergebnisse aus AP 2.2:

Neukonzeption des Datenmodells, basierend auf den ermittelten Mängeln:

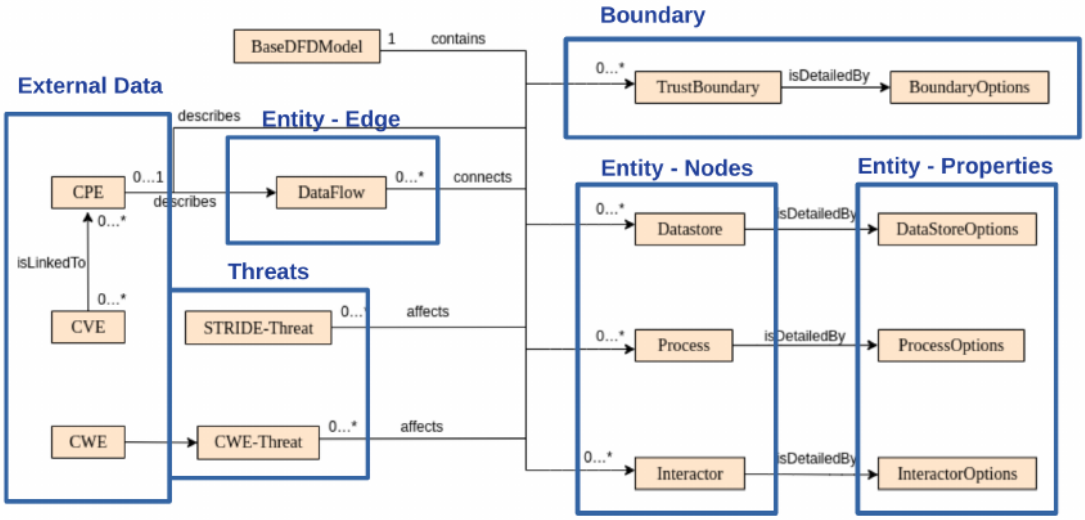


Abbildung 4: Abstraktion der Elemente des Datenmodells

Auf Basis der gruppierten Elemente des Datenmodells wurde der Kern des neuen Datenmodells definiert. Das Datenmodell dieses Applikationskerns soll die notwendige Grundstruktur für die spätere Implementierung der Erkennungslogik bereitstellen, welche anschließend für die tatsächliche Erkennung durch das Plugin-System erweitert wird. Es handelt sich somit um das notwendige Minimum, welches für jede Bedrohungsanalyse verwendet wird, ohne die konkreten domänenspezifischen Inhalte der Analysemethode zu betrachten."

Entwurf zur Neukonzeption mit Plugin-System:

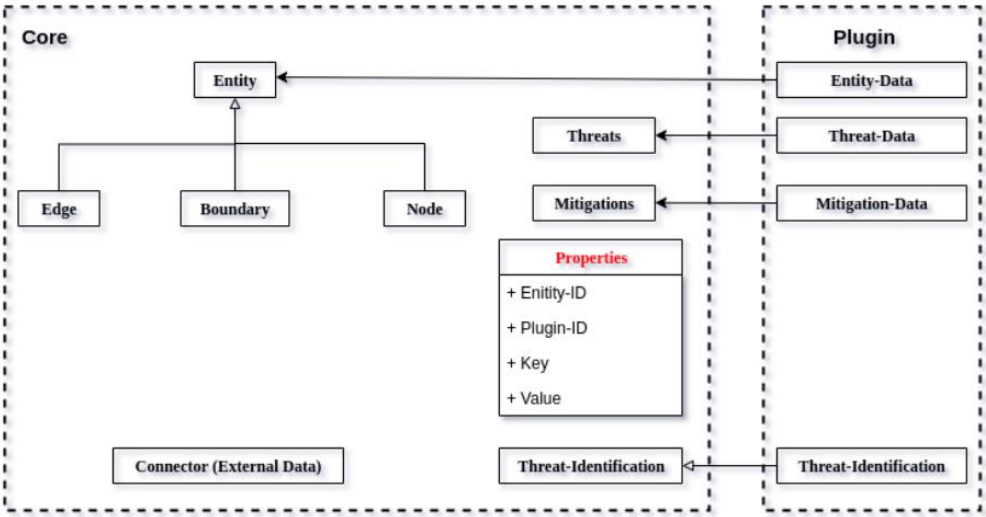


Abbildung 5: Kernelemente des neues Datenmodells

Als Übergang in die Komponentenmodellierung (Arbeitspaket 3) ist in Abbildung 7 die Interaktion der definierten Kern-Komponenten (abgeleitet aus den Datenstrukturen) dargestellt. Dabei fließen die Komponenten des DFD-Modells (Edges, Nodes, Boundaries) in die durch Plugins erweiterbare Threat-Identification-Engine ein. Je nach Plugin-Logik greift diese Komponente auf externe Datenquellen zurück, um Mitigations und Threats zu definieren, welche anschließend im Report-Element zusammengefasst werden.

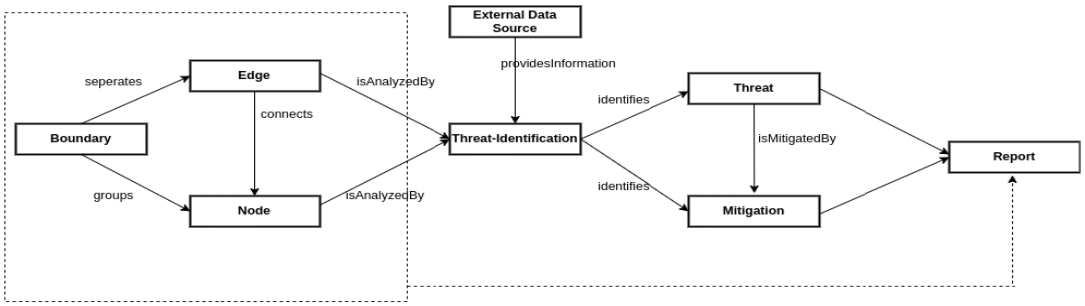
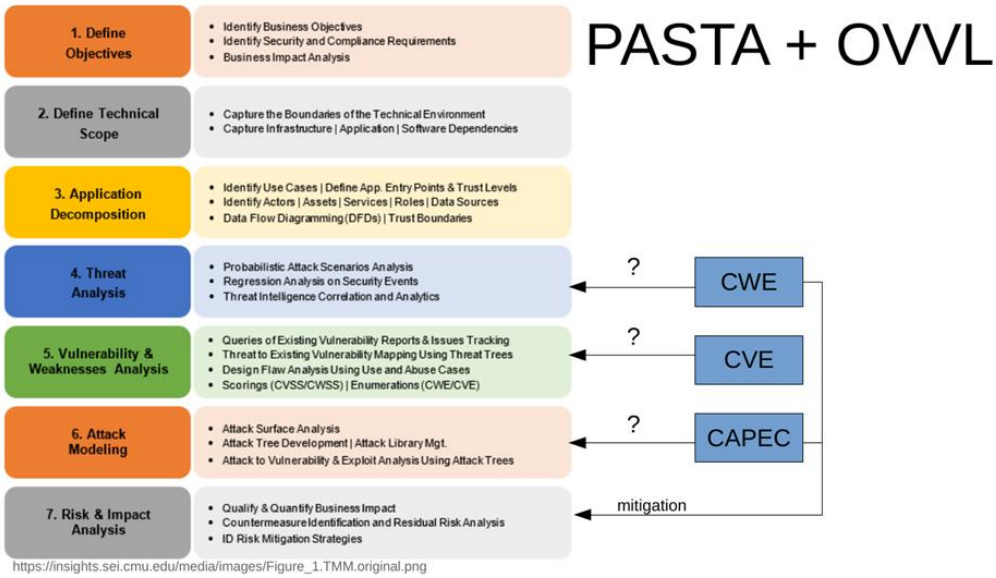


Abbildung 7: Übergang Komponentenmodell - Interaktion

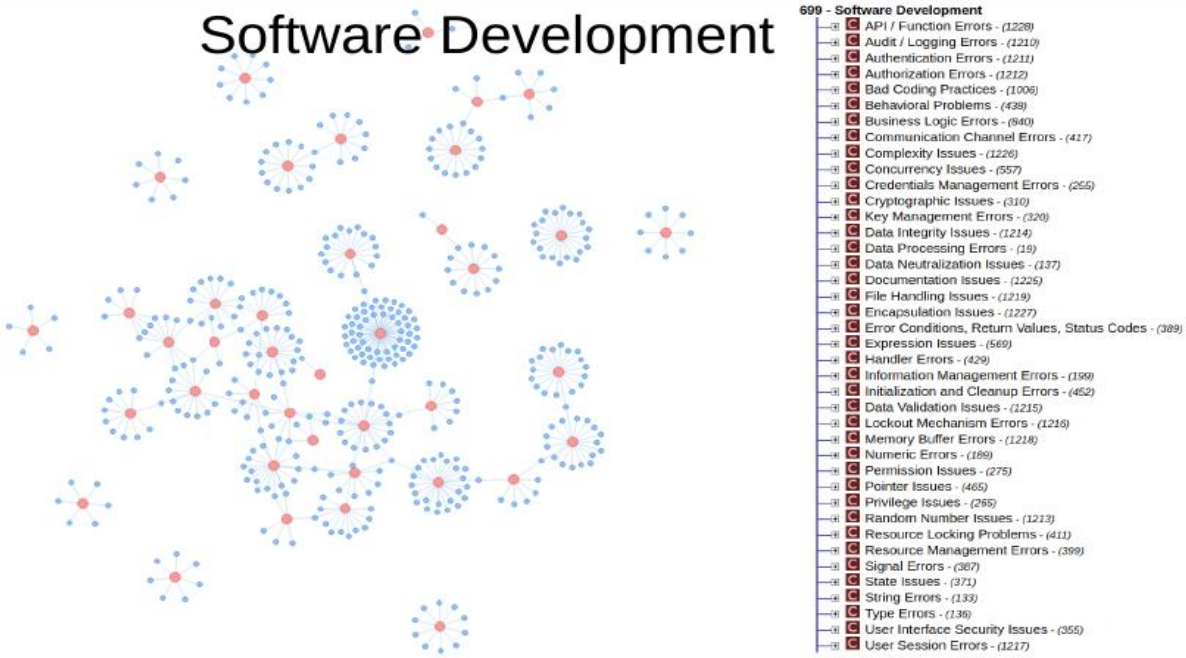
Fazit: Im Rahmen dieses Arbeitspakets sollte ermittelt werden, ob eine Neuentwicklung des Tools angemessen ist. Aufgrund der ermittelten Defizite und den tiefgreifenden Änderungen des Datenmodells und der Architektur wurde diese von den Projektbeteiligten beschlossen.

Das neue Datenmodell, welches speziell auf die Benutzung eines erweiterbaren Pluginsystems ausgelegt ist, wurde in AP2.2 definiert und in einem technischen Bericht festgehalten.

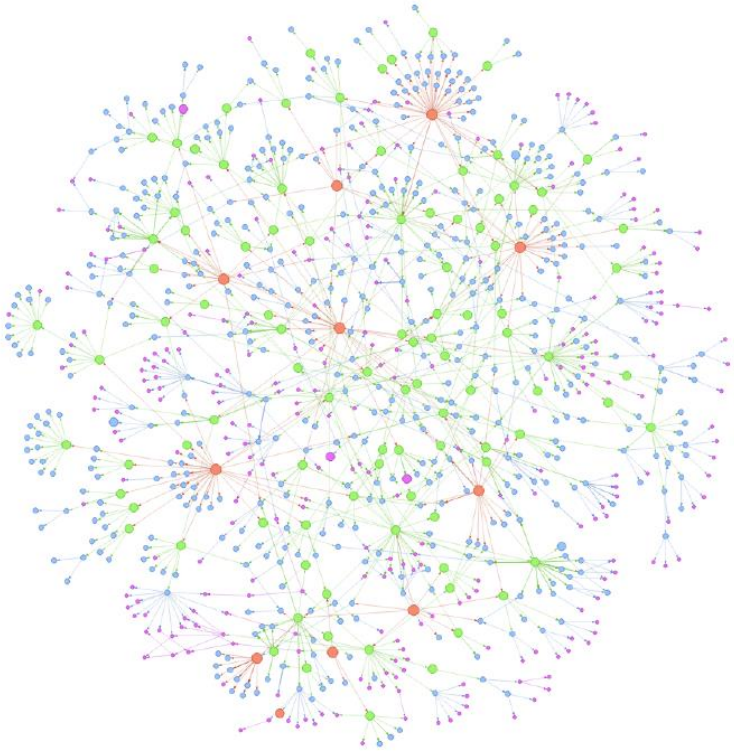
Zudem wurde Verwendung der Pasta Methodik und Anreicherung der Methodik mit externen Daten geprüft:



Der CWE Datensatz wurde analysiert, es wurde geprüft ob die Inhalte des Datensatzes (Schwächen/Weakness) und die Kategorisierung dieser, geeignet sind um in der OVVL Threat modelling-Methodik verwendet zu werden (als Schwachstellendatenbank); Ergebnis: nicht geeignet.



Research Concepts



Pillar: 9
Class: 97
Base: 518
Variant: 289

2.3 AP 3: Entwurf OVVL Komponentenmodell (16 PM)

Das Komponentenmodell (AP3) wurde entworfen und hingehend der Anforderungen geprüft. Dieses setzt sich aus der Kernanwendung (umgesetzt als Spring Boot Applikation) und den, durch Drittparteien erweiterbaren Plugins (umgesetzt durch das PF4J-Framework) zusammen.

Im Rahmen des AP3 (Entwurf OVVL Komponentenmodell) hat die Hochschule Offenburg die Entwicklung der OVVL-Applikation, basierend auf dem im Vorjahr entworfenen Komponentenmodell, durchgeführt. Die serverseitige Hauptanwendung wurde wie geplant als Spring-Boot-Applikation implementiert. Dabei wurde das PF4J-Framework genutzt, um die Anwendung durch Plugins erweitern zu können.

Die Modulimplementierung wird parallel mit den Projektpartnern bearbeitet. In Zusammenarbeit mit den Partnern der ENRW wurde die Entwicklung des DSGVO-Moduls diskutiert, wobei die LINDDUN Methodik als Grundlage diente. Gleichzeitig prüfen die Projektpartner der Secuvera die KI-gestützte Analyse.

Die Inhalte des AP3 wurden in einer umfangreichen Dokumentation festgehalten. Diese Dokumentation umfasst neben einer detaillierten Beschreibung des Aufbaus und der Funktionsweise der Anwendung auch eine Anleitung zur Entwicklung neuer Plugins. Zusätzlich enthält sie praktische Beispiele zur Verwendung der OVVL API und zur Integration von OVVL in Drittanwendungen."

Übersicht:

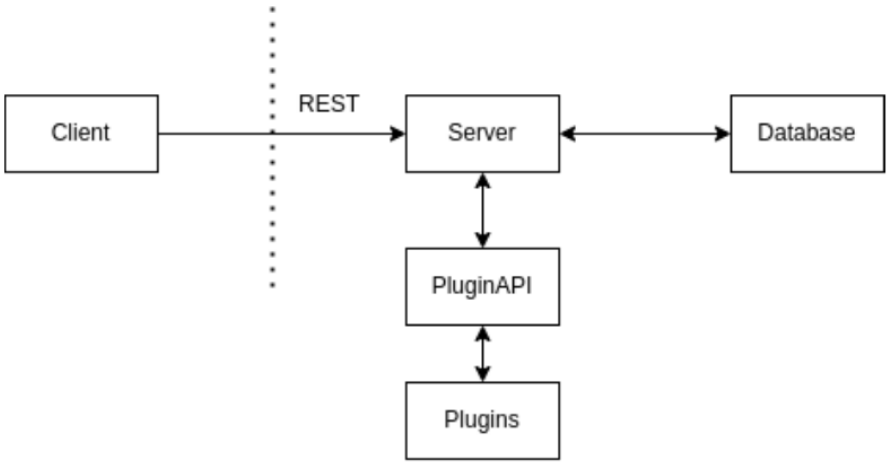
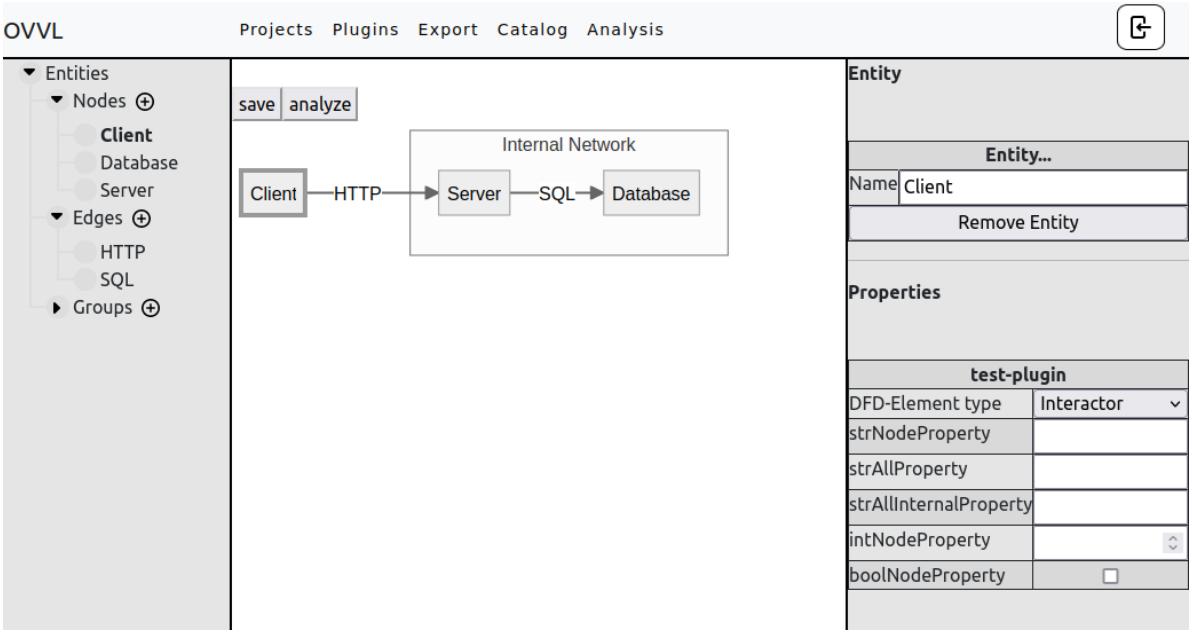


Abbildung 1: Übersicht: Architektur

Im Austausch mit den Projektpartnern wurde deutlich, dass ein Bedarf für einen Client zur praktischen Nutzung der Anwendung besteht. Daraufhin wurde der Umfang der geplanten Entwicklung erweitert und die Mitarbeiter der HSO haben eine clientseitige Anwendung mit einer benutzerfreundlichen Benutzeroberfläche implementiert.

Der Client wurde als Webanwendung mithilfe von Angular entwickelt, was es dem Endnutzer ermöglicht, die Anwendung plattformunabhängig im Browser zu nutzen.

Erster Proof of Concept



Neuer Proof of Concept

Ausführliche Screenshots siehe Anhang

Arbeiten und Ergebnisse aus AP 3.1: Attack Tree Modul

Es wurde untersucht wie Attack Trees in Mittels der Meta Attack Language (MAL) in OVVL genutzt werden können. Hierfür wurden existierende Ansätze betrachtet (https://link.springer.com/article/10.1007/s10270-021-00898-7/tables/1), die zeigen wie eine domänenspezifische, auf MAL basierende Sprache genutzt werden kann, um aus dem MITRE ATT&CK Datensatz Angriffsbäume zu generieren, die in unserer Anwendung verwendet werden können. Die Analyse ergab, dass das von uns erstellte Pluginmodell eine solche Funktion grundsätzlich erfüllen könnte.

(Ansatz wurde in OVVL nicht implementiert, POC für MAL in Python wurde veröffentlicht: https://github.com/victorazzam/demal/)

Der aktuelle Forschungsstand im Bereich der Attack-Trees wurde untersucht und die Anforderungen einer Implementierung dieser Analysemethode wurde mit dem erstellten Datenmodell abgeglichen. Aus diesem Abgleich ergab sich, dass die Implementierung einer Attack-Tree-basierten Analyse durch das genutzte Pluginmodell möglich ist. Des Weiteren wird gegenwärtig untersucht, ob aufbauend auf aktuellen Forschungsarbeiten eine Kombination einer automatisierten KI-gestützen Attack-Tree Generierung umgesetzt werden kann.

Arbeiten und Ergebnisse aus AP 3.2: Identifikation von Datenflüssen

Implizit durch Portierung von alter OVVL Logik. Generell lassen sich mit dem Tool Datenflüsse detailliert Modellierung und bei der Identifikation von Bedrohungen ist es beispielsweise auch möglich, Datenflüsse über mehrere Knotenpunkte hinweg zu modellieren (beispielsweise Angriffspfade von einem initialen Angriffspunkt bis zu einem Zielsystem zu analysieren). Existierende Arbeiten hierzu wurden auch genauer untersucht.

Wir haben die Programmlogik des alten OVVL Tools als Plugin umgesetzt, bei dem Datenflüsse jeweils im Kontext ihres Ursprungs und der Quelle betrachtet werden. Auf Basis der Bedrohungen des “Elevation of Privilege (EoP) Threat Modeling Card Game”. Dabei werden statische Regeln der kommunizierenden Knoten geprüft, um Bedrohungen zu identifizieren. Die Methodik wurde außerdem licht überarbeitet, sodass die Anzahl der verwendeten Attribute im Vergleich zur alten Implementierung reduziert wurde. (Es werden nur noch Attribute gesetzt, die tatsächlich für die Identifikation benötigt werden)

Arbeiten und Ergebnisse aus AP 3.3: KI-gestütze Threat Analyse

KI-gestützte Analyse wurde geprüft, zu wenig brauchbare Datensätze, um ein sinnvolles Modell zu erstellen. Stattdessen wurde ein Plugin für die Bedrohungsmodellierung für KI-Systeme erstellt (SMILE-Threatkatalog). Funktionsweise des erstellten Moduls:

- Prozessarten aus dem Machine Learning Lifecycle werden im Modell zugewiesen ("Training", "Pre-processing", "Inference", "Data Collection", "Data Storage", "Model Deployment", "Model Monitoring")
- Arten von Daten werden zugewiesen an Prozesse und Datenflüsse ("Training Data", "Model Data", "Inference Data", "Metadata")
- Sensitivität und Verschlüsselung von Daten werden zugewiesen

Die gesetzten Attribute werden mit den festgelegten Regeln unserer Bedrohungsbibliothek abgeglichen, Bedrohungen werden dadurch ermittelt.

Arbeiten und Ergebnisse aus AP 3.4: DSGVO / GDPR

Gemeinsam mit ERNW wurde LINDDUN als OVVL Plugin umgesetzt, wird mit veröffentlicht. Methodik: In der OVVL Applikation werden LINDDUN Attribute genutzt um Privacy Threats zu erkennen (Funktionsweise ähnlich wie ein Fragebogen)

2.4 AP 4: Verwendung von OVVL in Sicherheitsprozessen (0 PM)

kein geplanter Aufwand für die HSO

2.5 AP 5: Entwurf Prozess- und Betriebsmodell (0 PM)

kein geplanter Aufwand für die HSO

2.6 AP 6: Entwurf OVVL Plattform & Integration (6 PM)

Als ersten Schritt zur plattformunabhängigen Bereitstellung der OVVL-Applikation (AP 6.1 - Bereitstellung als Cloud-Dienst) wurde beschlossen, eine Docker-basierte Umgebung bereitzustellen. Diese Umgebung erfüllt die Kriterien für reproduzierbares Deployment und ermöglicht eine modulare Einbindung der Plugins.

Die Inhalte des AP 6.2 (Einbindung individueller Module) sind implizit durch die Verwendung des Plugin-Systems bereits erfüllt. Das System ermöglicht die Einbindung, Entwicklung und flexible Verwendung beliebiger Module.

Eine Docker basierte Umgebung wurde eingerichtet, wurde gemeinsam mit dem restlichen Code veröffentlicht.

Ein Deploy Prozess wurde in der bw-cloud durchgeführt und für eine einfache Replikation dokumentiert. Der Prozess umfasst:

- Einrichtung der Cloud-Instanz
- Einrichtung des SSH-Zugriffs
- Verwendete Docker Files
- Nutzung von Certbot und Erstellung der Letsencrypt Zertifikate
- Nginx Konfiguration

Nutzung der OVVL Deployment Skripte, um die Docker Umgebung zu starten

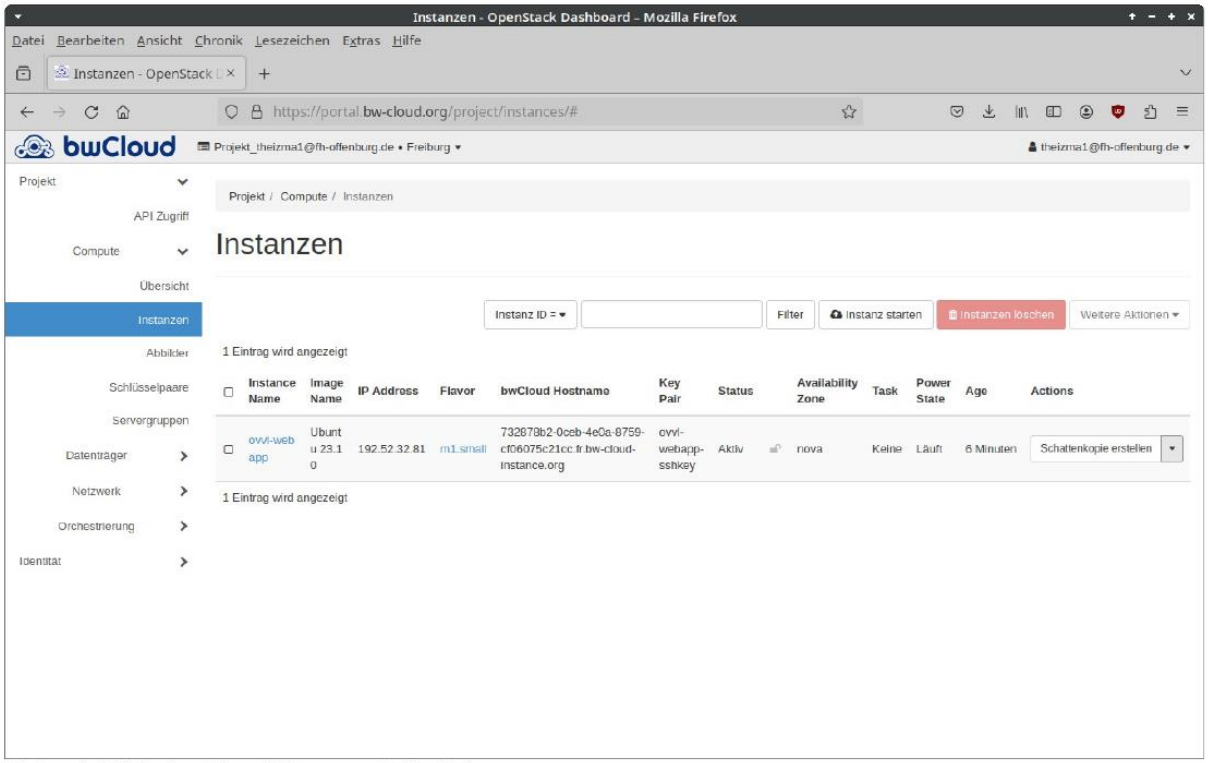


Schaubild 1: bwCloud Instanz Beispiel

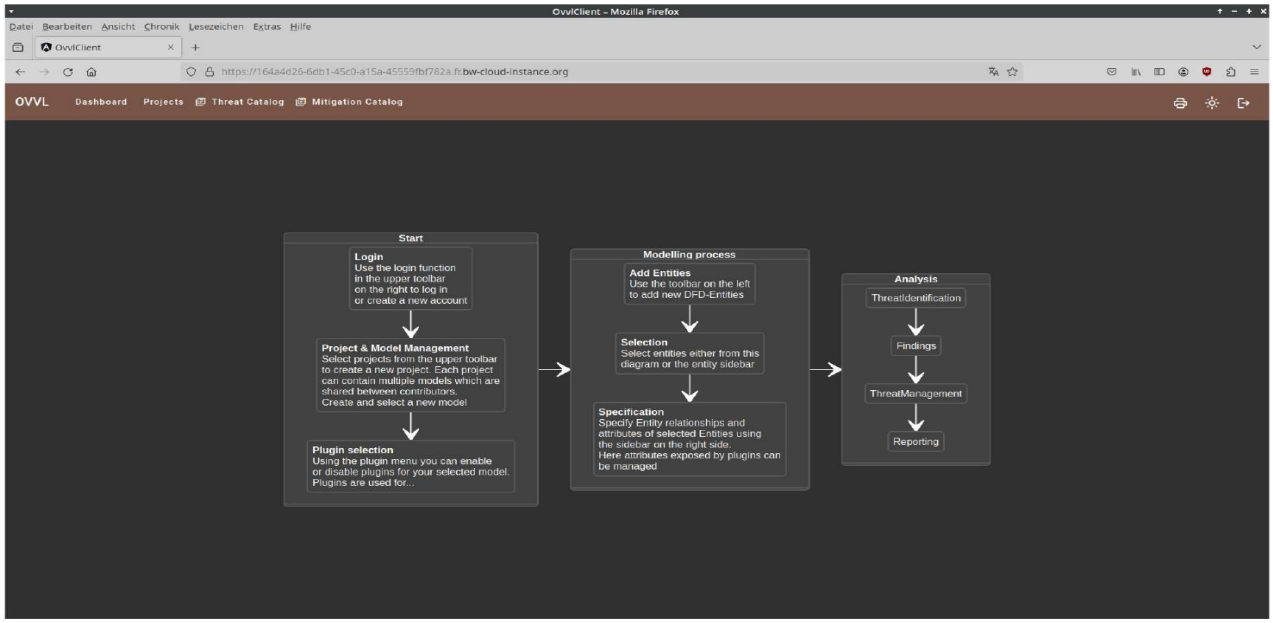


Abbildung 6: OVVL Anwendung in der Cloud

2.7 AP 7: Validierung, Test und Evaluation (6 PM)

Um den Austausch zwischen den Projektpartnern über den Umsetzungsgrad der Anforderungen zu erleichtern, wurden Teile des AP7 (Validierung, Test und Evaluation) bereits vorgezogen. In Zusammenarbeit mit der ERNW wurde ein Dokument erstellt, das die erfassten Anforderungen mit dem aktuellen Entwicklungsstand abgleicht.

Das Dokument enthält zusätzlich technische Details, die zeigen, wie die Anforderungen konkret umgesetzt wurden, sowie Informationen darüber, welche Anforderungen vollständig oder teilweise nicht im Umfang der Entwicklung umgesetzt wurden.

Codetests: Für Kernelemente der implementierten Anwendung wurde die Funktionalität durch Unit- und Integrationstests geprüft.

3 Vergleich des Vorhabensstands mit der ursprünglichen (bzw. mit Zustimmung des Zuwendungsgebers geänderten) Arbeits-, Zeit- und Ausgabenplanung

Es gab keine relevanten Abweichungen in Arbeits-, Zeit- und Ausgabenplanung, alle Ziele wurden erreicht.

4 Fortschreibung des Verwertungsplans

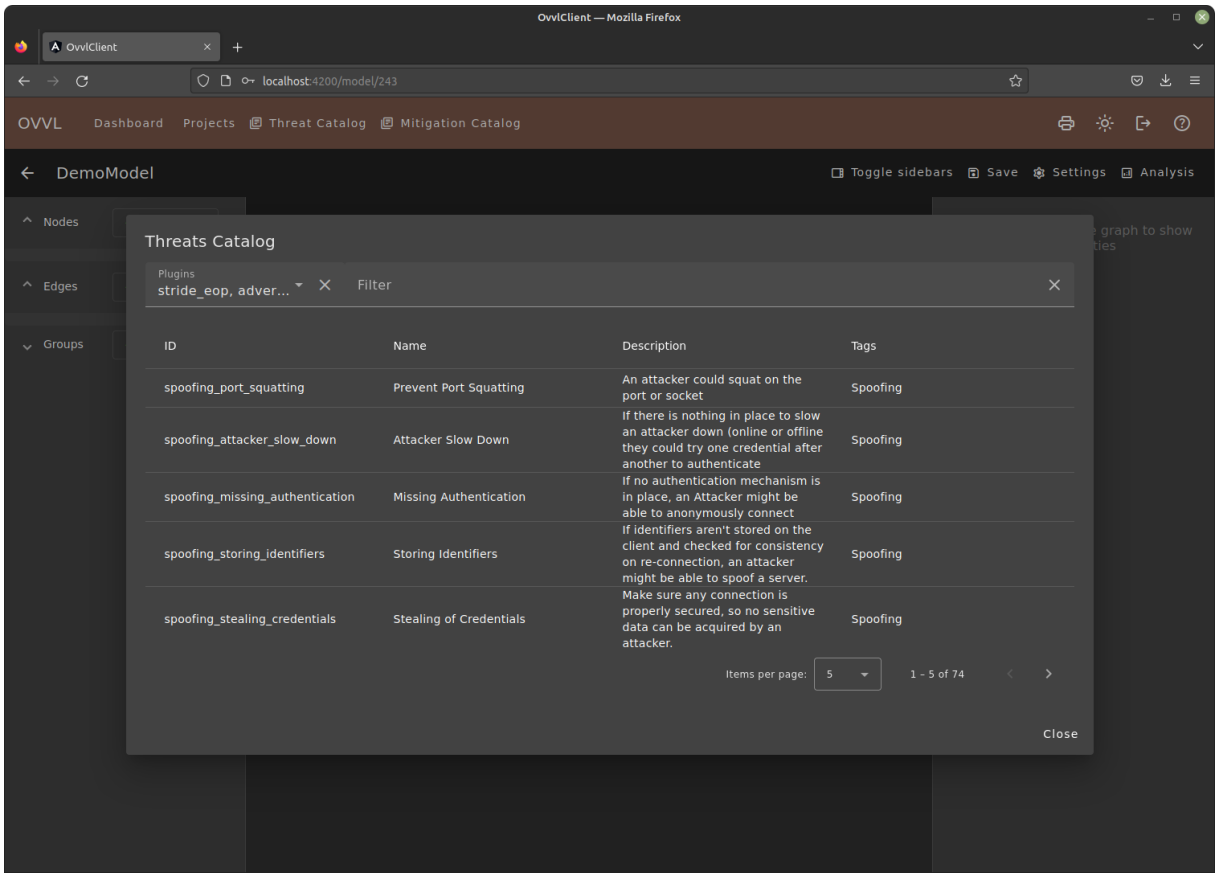
4.1 Wissenschaftliche und/oder technische Erfolgsaussichten nach Projektende (mit Zeithorizont)

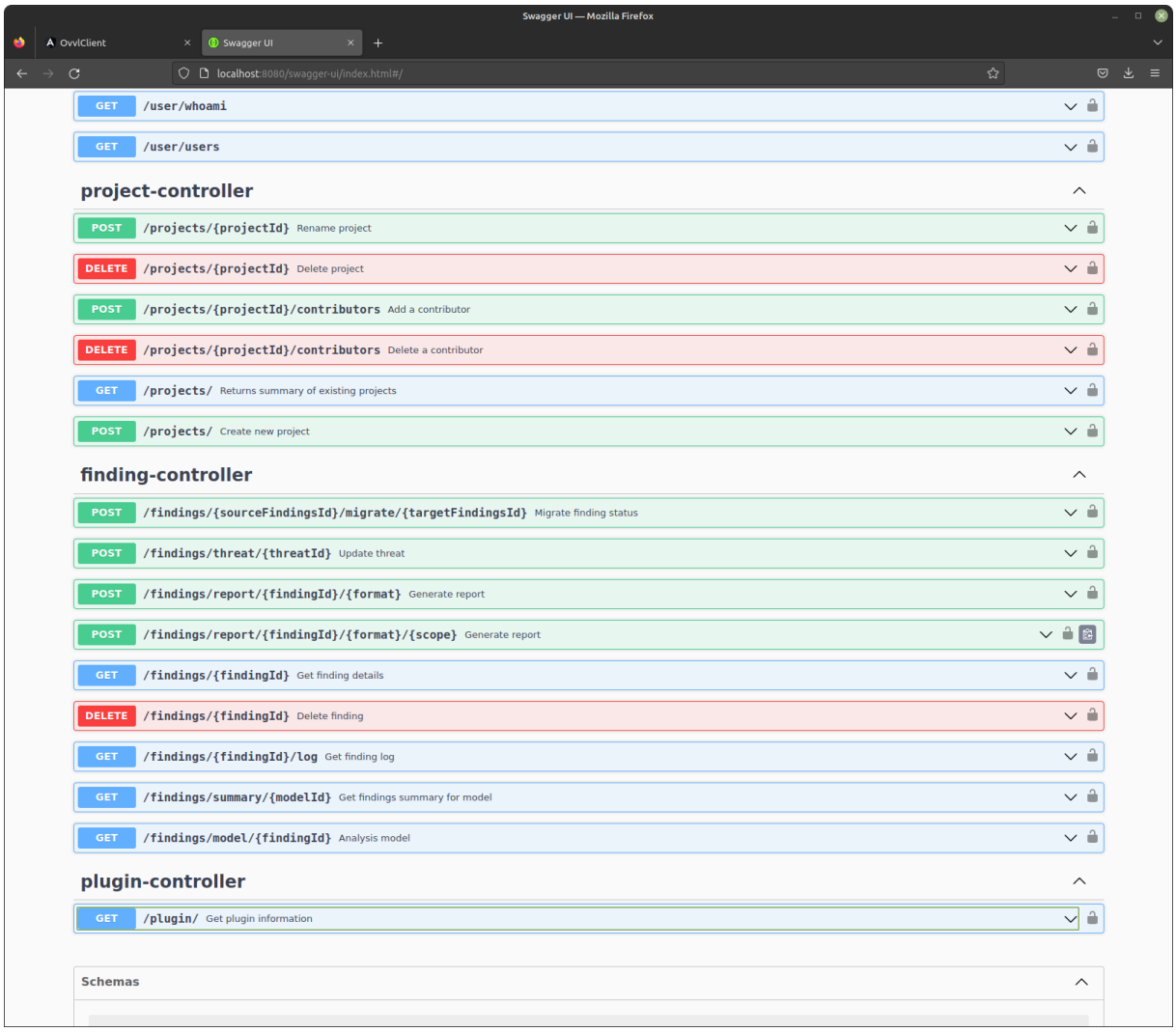
Ein in Teilen auf OVVL basierendes Folgeprojekt (primär basierend auf AP3.4: KI-basierte Threat Analyse) wurde durch die HSO erfolgreich initiiert (3 jähriges KMU Innovativ Projekt „SMILE“ FKZ 16KIS1929, 01.08.2023-31.07.2026).

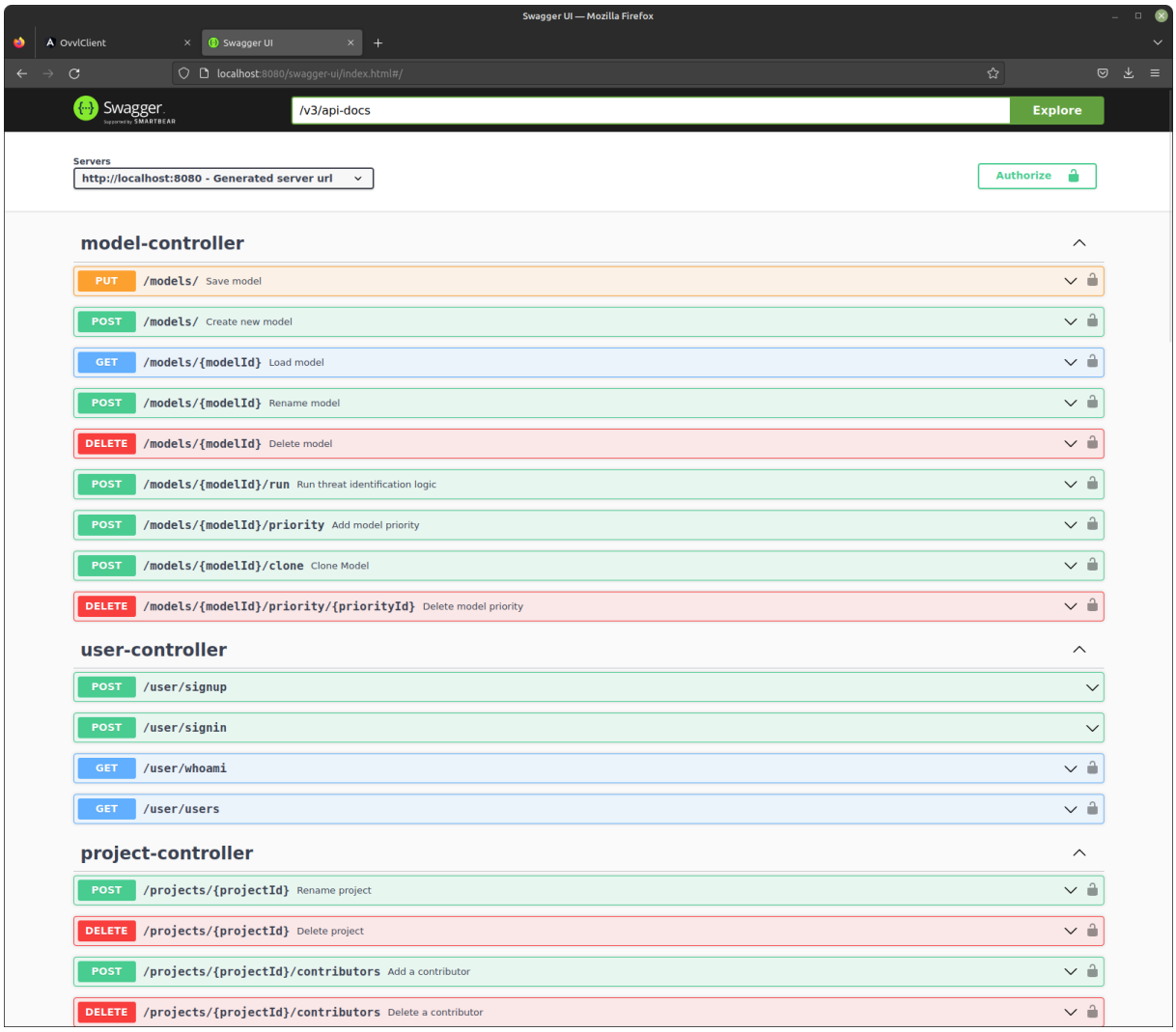
4.2 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit für eine mögliche notwendige nächste Phase bzw. die nächsten innovatorischen Schritte zur erfolgreichen Umsetzung der Ergebnisse.

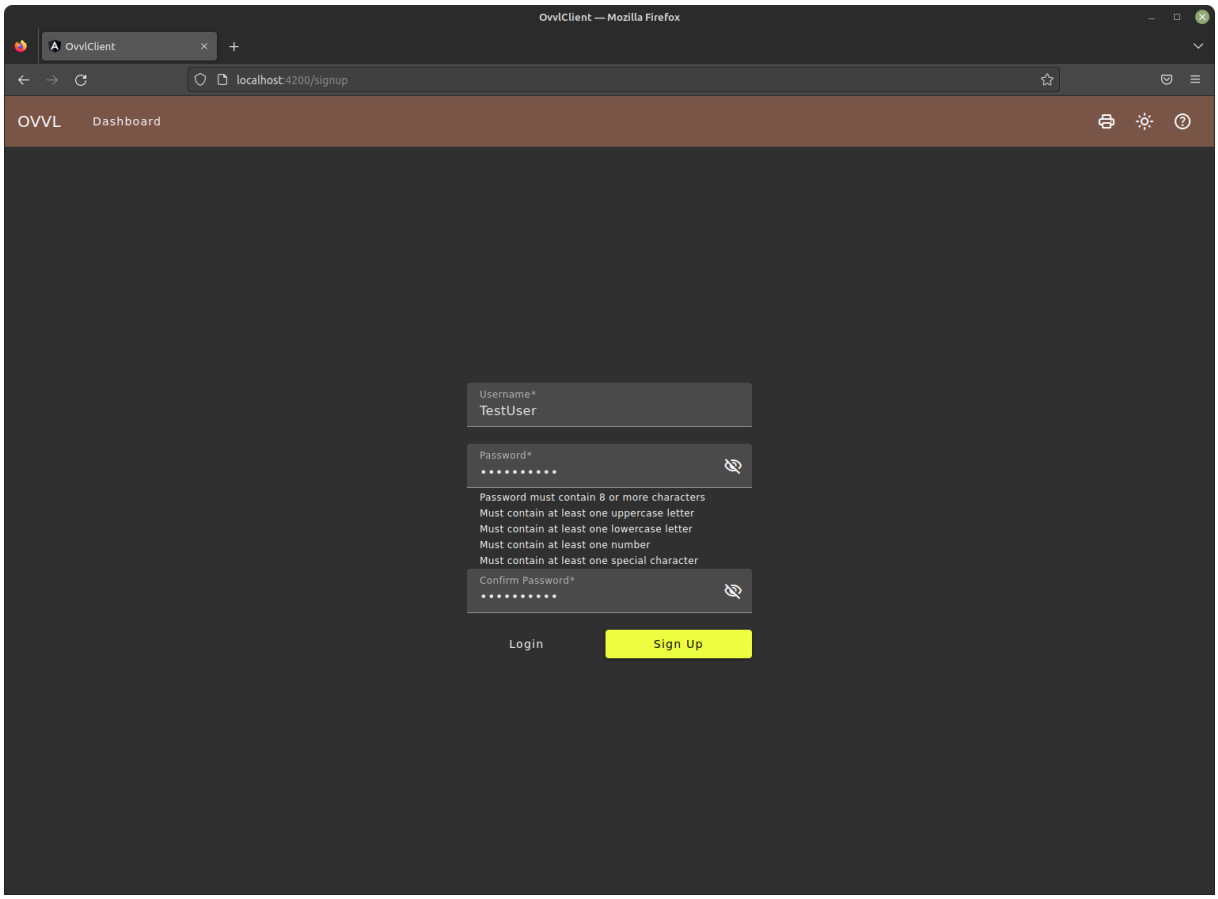
Die Resultate und erstelltes Tool (OVVL) werden von Prof. Schaad in den Master Studiengang (ENITS) der HSO und der Vorlesung „Software Security“ verwendet.

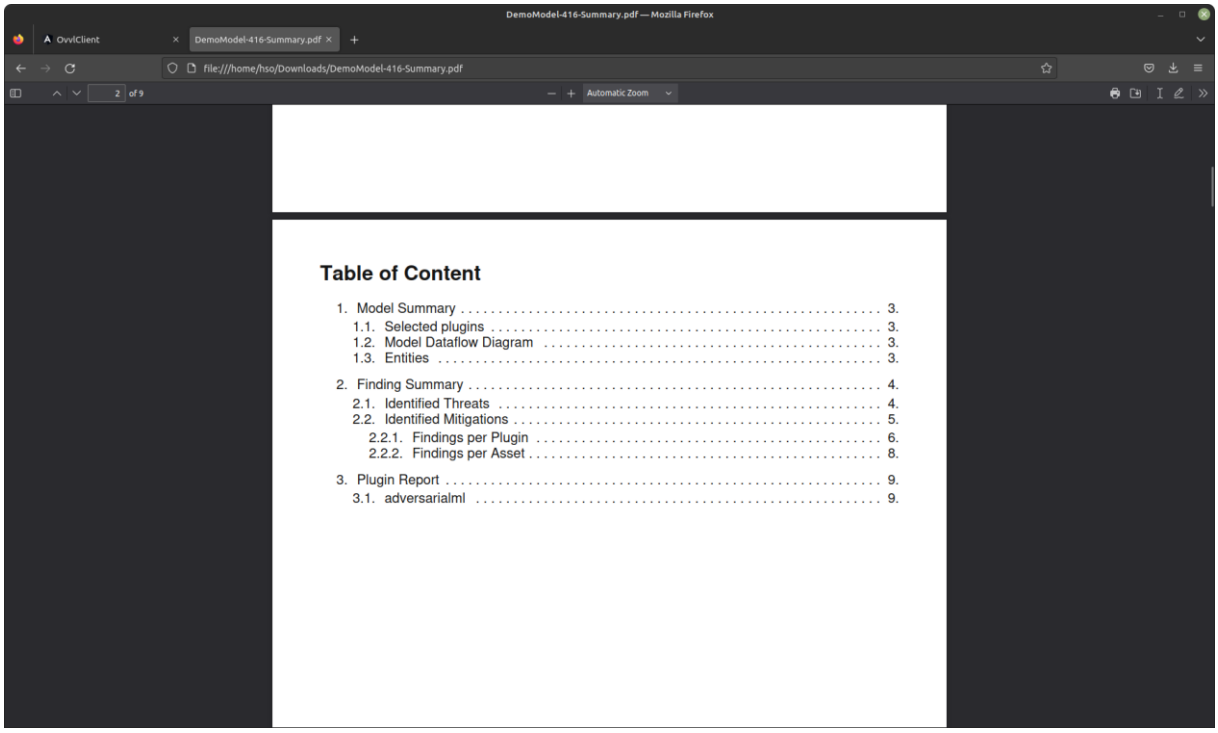
5 Anhang: Screenshots OVVL Proof of Concept

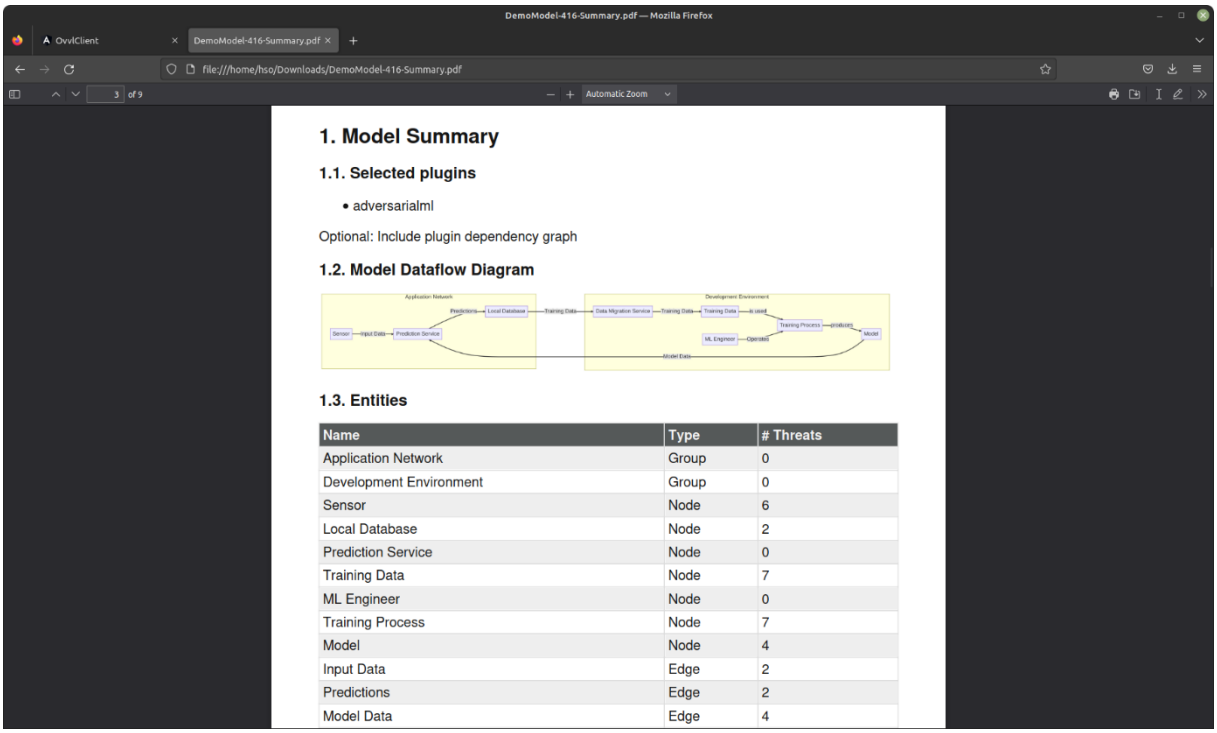

OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/model/243
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoModel
Toggle sidebars
Save
Settings
Analysis
Nodes
Edges
Groups
Threats Catalog
Plugins
stride_eop, adver...
Filter
ID
Name
Description
Tags
spoofing_port_squatting
Prevent Port Squatting
An attacker could squat on the port or socket
Spoofing
spoofing_attacker_slow_down
Attacker Slow Down
If there is nothing in place to slow an attacker down (online or offline they could try one credential after another to authenticate
Spoofing
spoofing_missing_authentication
Missing Authentication
If no authentication mechanism is in place, an Attacker might be able to anonymously connect
Spoofing
spoofing_storing_identifiers
Storing Identifiers
If identifiers aren't stored on the client and checked for consistency on re-connection, an attacker might be able to spoof a server.
Spoofing
spoofing_stealing_credentials
Stealing of Credentials
Make sure any connection is properly secured, so no sensitive data can be acquired by an attacker.
Spoofing
Items per page:
5
1 - 5 of 74
Close


Swagger UI — Mozilla Firefox
OvvlClient
Swagger UI
localhost:8080/swagger-ui/index.html#/
GET /user/whoami
GET /user/users
project-controller
POST /projects/{projectId} Rename project
DELETE /projects/{projectId} Delete project
POST /projects/{projectId}/contributors Add a contributor
DELETE /projects/{projectId}/contributors Delete a contributor
GET /projects/ Returns summary of existing projects
POST /projects/ Create new project
finding-controller
POST /findings/{sourceFindingsId}/migrate/{targetFindingsId} Migrate finding status
POST /findings/threat/{threatId} Update threat
POST /findings/report/{findingId}/{format} Generate report
POST /findings/report/{findingId}/{format}/{scope} Generate report
GET /findings/{findingId} Get finding details
DELETE /findings/{findingId} Delete finding
GET /findings/{findingId}/log Get finding log
GET /findings/summary/{modelId} Get findings summary for model
GET /findings/model/{findingId} Analysis model
plugin-controller
GET /plugin/ Get plugin information
Schemas


Swagger UI — Mozilla Firefox
OvvlClient
Swagger UI
localhost:8080/swagger-ui/index.html#/
Swagger
/v3/api-docs
Explore
Servers
http://localhost:8080 - Generated server url
Authorize
model-controller
PUT /models/ Save model
POST /models/ Create new model
GET /models/{modelId} Load model
POST /models/{modelId} Rename model
DELETE /models/{modelId} Delete model
POST /models/{modelId}/run Run threat identification logic
POST /models/{modelId}/priority Add model priority
POST /models/{modelId}/clone Clone Model
DELETE /models/{modelId}/priority/{priorityId} Delete model priority
user-controller
POST /user/signup
POST /user/signin
GET /user/whoami
GET /user/users
project-controller
POST /projects/{projectId} Rename project
DELETE /projects/{projectId} Delete project
POST /projects/{projectId}/contributors Add a contributor
DELETE /projects/{projectId}/contributors Delete a contributor


OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/signup
OVVL
Dashboard
Username*
TestUser
Password*
Password must contain 8 or more characters
Must contain at least one uppercase letter
Must contain at least one lowercase letter
Must contain at least one number
Must contain at least one special character
Confirm Password*
Login
Sign Up


DemoModel-416-Summary.pdf — Mozilla Firefox
OvvlClient
DemoModel-416-Summary.pdf
file:///home/hso/Downloads/DemoModel-416-Summary.pdf
2 of 9
Automatic Zoom
Table of Content
1. Model Summary . . . . . 3.
1.1. Selected plugins . . . . . 3.
1.2. Model Dataflow Diagram . . . . . 3.
1.3. Entities . . . . . 3.
2. Finding Summary . . . . . 4.
2.1. Identified Threats . . . . . 4.
2.2. Identified Mitigations . . . . . 5.
2.2.1. Findings per Plugin . . . . . 6.
2.2.2. Findings per Asset . . . . . 8.
3. Plugin Report . . . . . 9.
3.1. adversarialml . . . . . 9.



1. Model Summary

1.1. Selected plugins

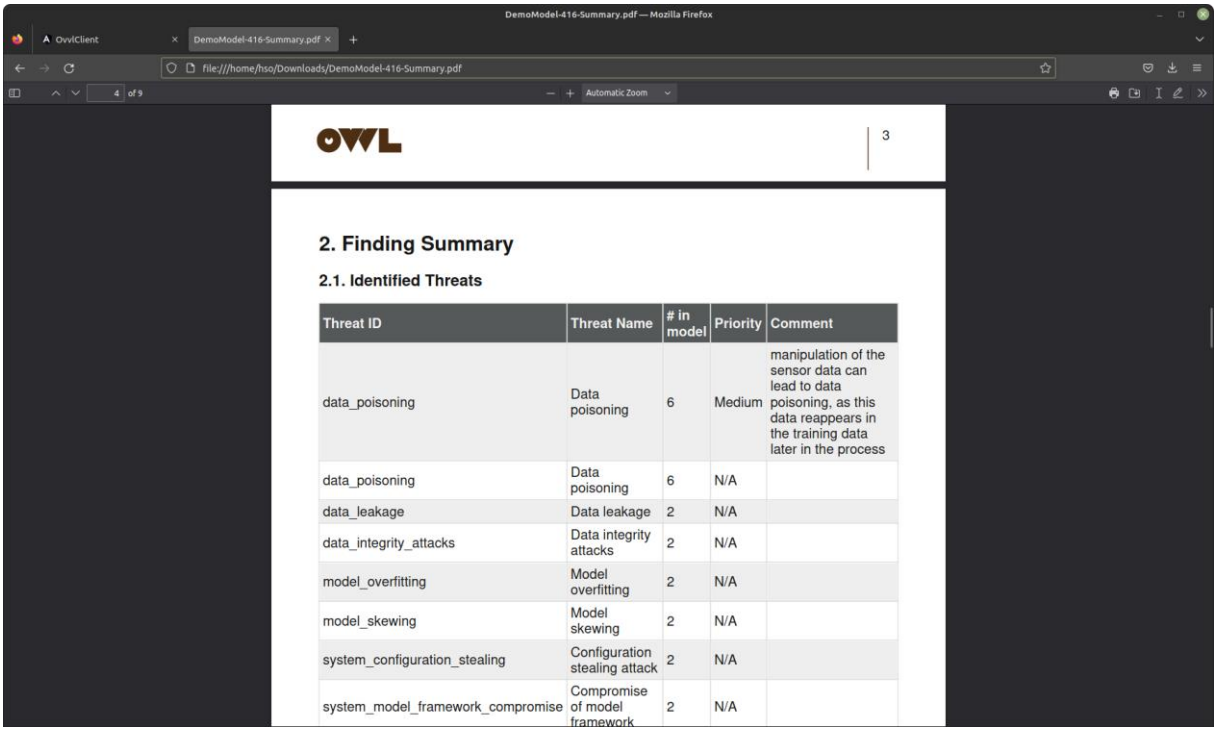
- adversarialml

Optional: Include plugin dependency graph

1.2. Model Dataflow Diagram

1.3. Entities

Name	Type	# Threats
Application Network	Group	0
Development Environment	Group	0
Sensor	Node	6
Local Database	Node	2
Prediction Service	Node	0
Training Data	Node	7
ML Engineer	Node	0
Training Process	Node	7
Model	Node	4
Input Data	Edge	2
Predictions	Edge	2
Model Data	Edge	4

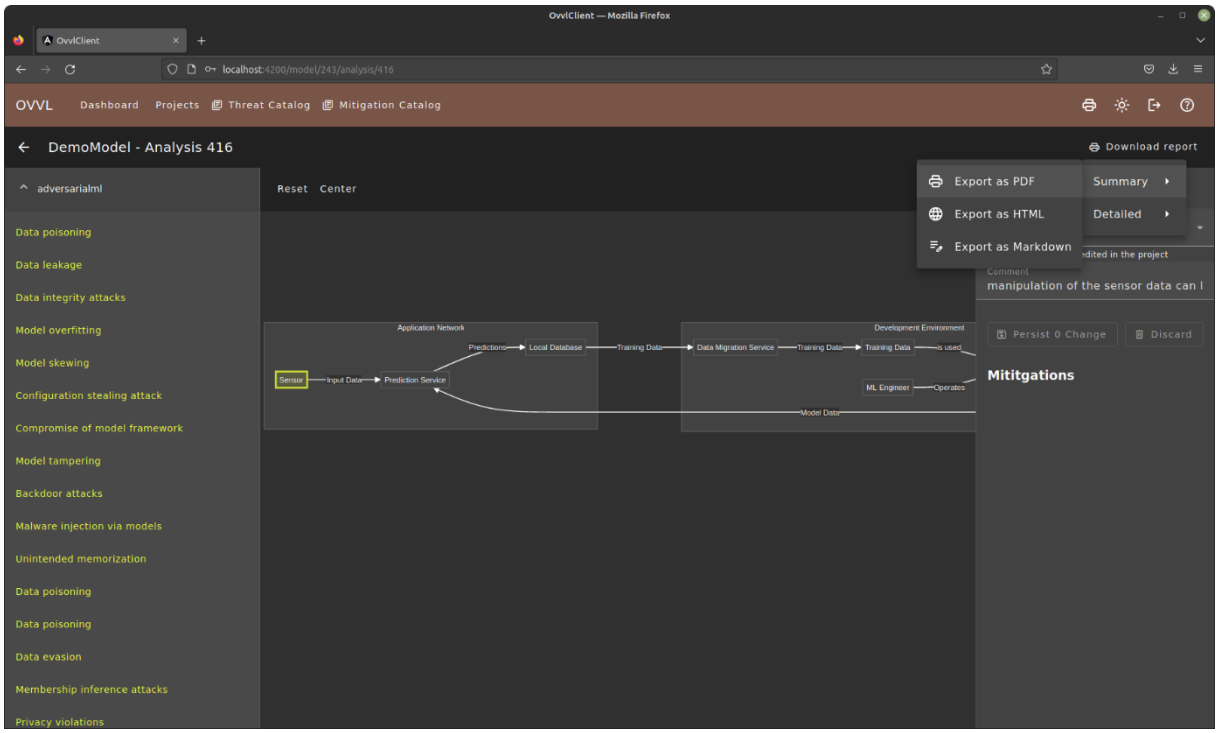


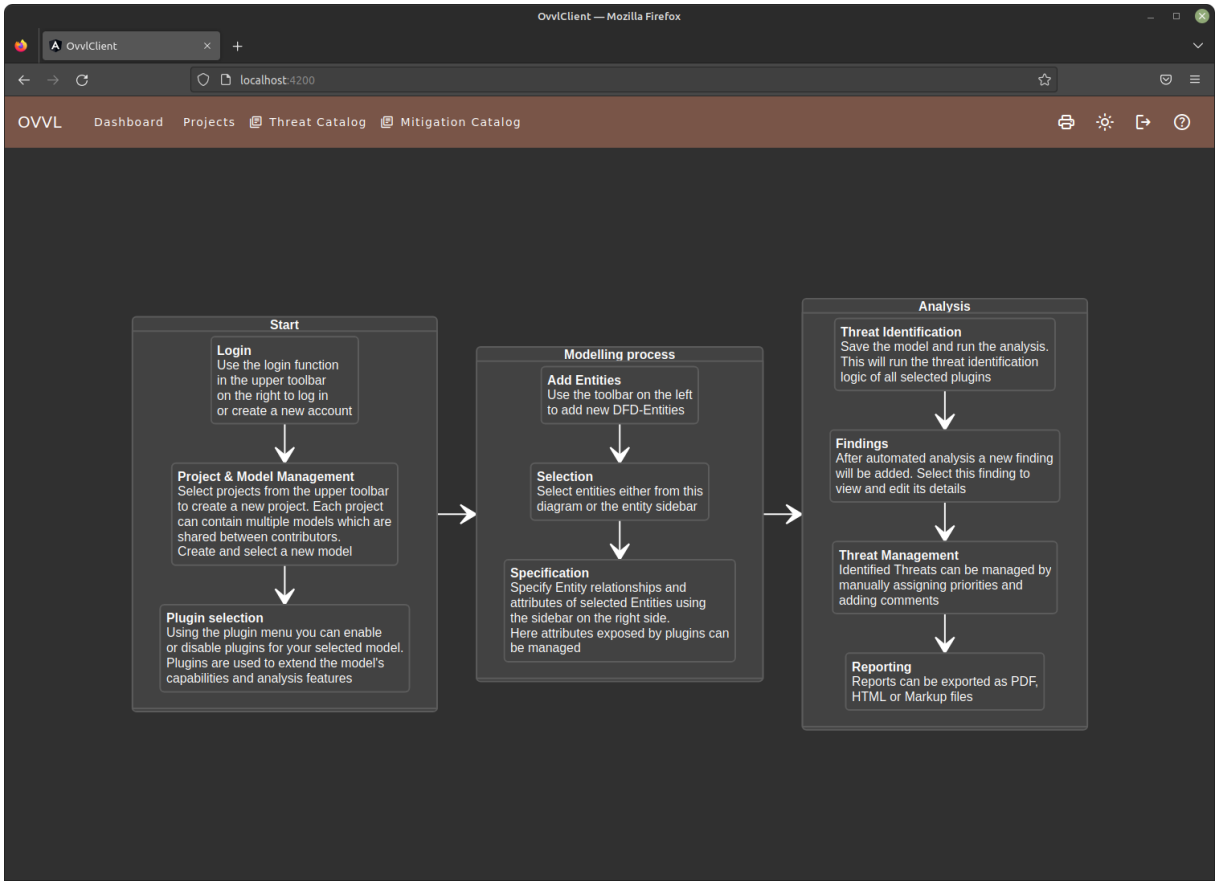
3

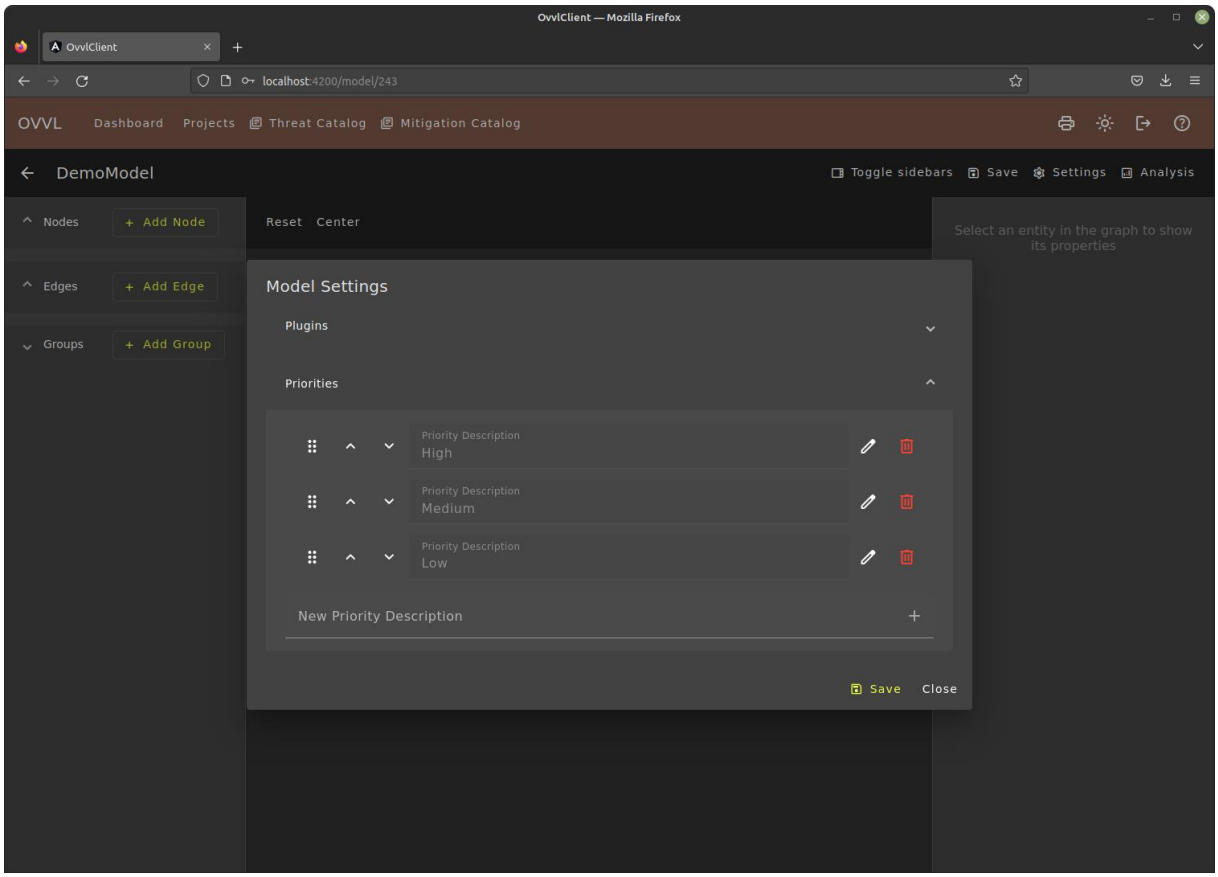
2. Finding Summary

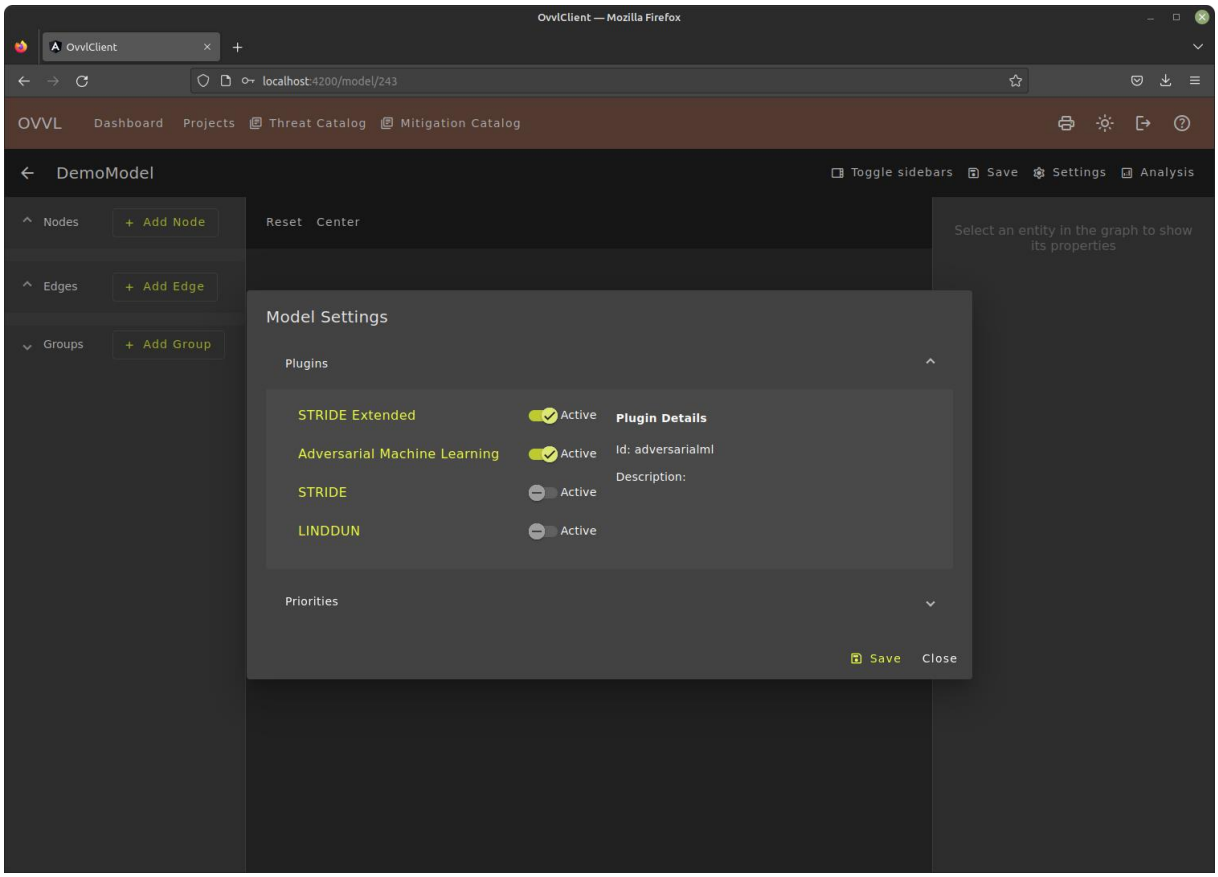
2.1. Identified Threats

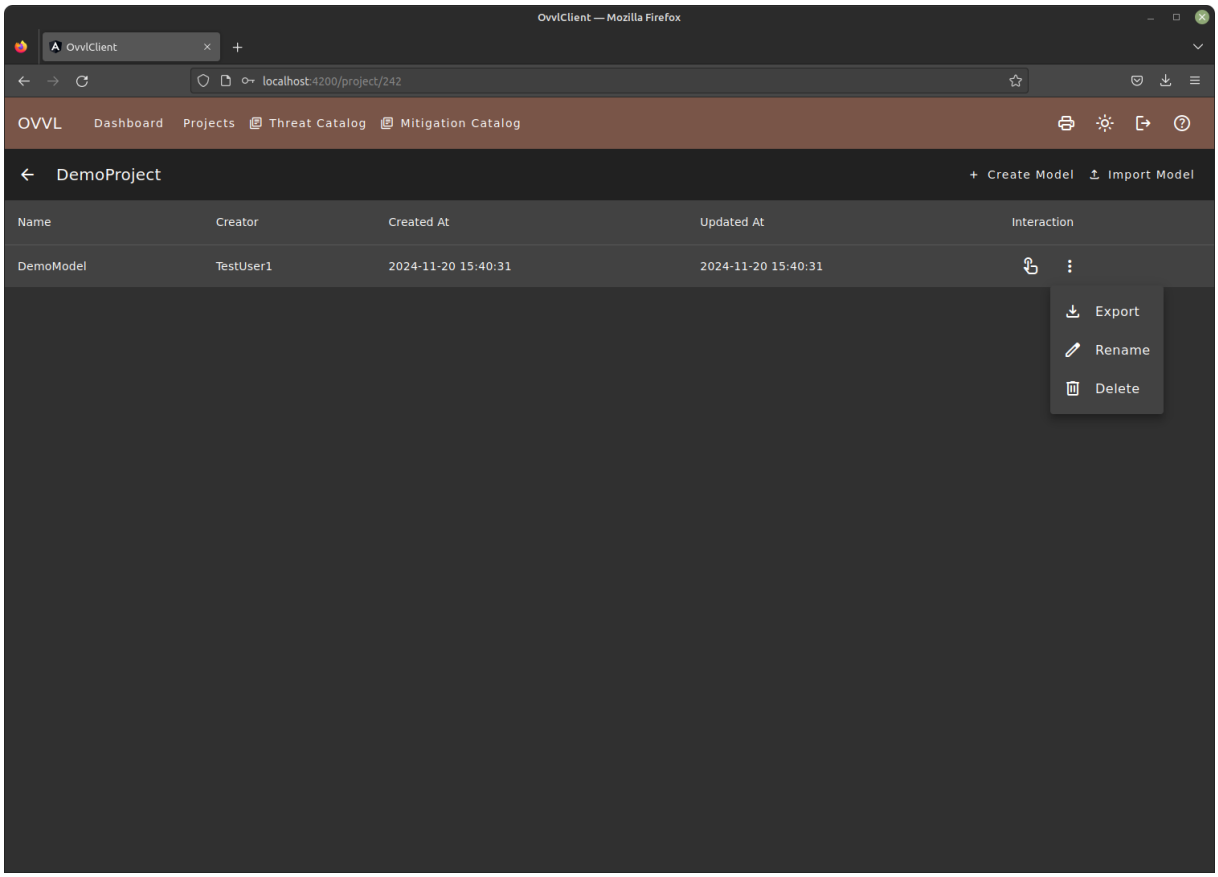
Threat ID	Threat Name	# in model	Priority	Comment
data_poisoning	Data poisoning	6	Medium	manipulation of the sensor data can lead to data poisoning, as this data reappears in the training data later in the process
data_poisoning	Data poisoning	6	N/A	
data_leakage	Data leakage	2	N/A	
data_integrity_attacks	Data integrity attacks	2	N/A	
model_overfitting	Model overfitting	2	N/A	
model_skewing	Model skewing	2	N/A	
system_configuration_stealing	Configuration stealing attack	2	N/A	
system_model_framework_compromise	Compromise of model framework	2	N/A	

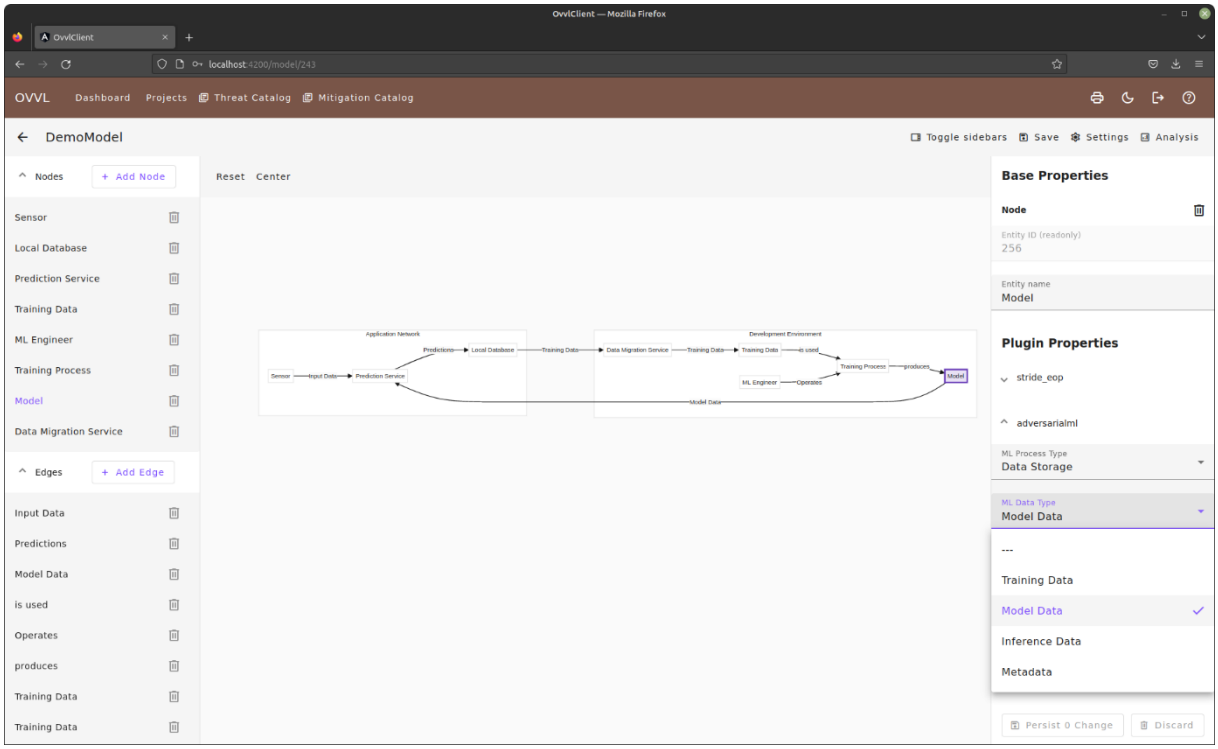

OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/model/243/analysis/416
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoModel - Analysis 416
Download report
adversarialml
Reset
Center
Export as PDF
Export as HTML
Export as Markdown
Summary
Detailed
Data poisoning
Data leakage
Data integrity attacks
Model overfitting
Model skewing
Configuration stealing attack
Compromise of model framework
Model tampering
Backdoor attacks
Malware injection via models
Unintended memorization
Data poisoning
Data poisoning
Data evasion
Membership inference attacks
Privacy violations
Comment
manipulation of the sensor data can l
Persist 0 Change
Discard
Mititgations
Application Network
Development Environment
Sensor
Input Data
Prediction Service
Predictions
Local Database
Training Data
Data Migration Service
Training Data
Training Data
is used
ML Engineer
Operates
Model Data


OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
Start
Login
Use the login function in the upper toolbar on the right to log in or create a new account
Project & Model Management
Select projects from the upper toolbar to create a new project. Each project can contain multiple models which are shared between contributors. Create and select a new model
Plugin selection
Using the plugin menu you can enable or disable plugins for your selected model. Plugins are used to extend the model's capabilities and analysis features
Modelling process
Add Entities
Use the toolbar on the left to add new DFD-Entities
Selection
Select entities either from this diagram or the entity sidebar
Specification
Specify Entity relationships and attributes of selected Entities using the sidebar on the right side. Here attributes exposed by plugins can be managed
Analysis
Threat Identification
Save the model and run the analysis. This will run the threat identification logic of all selected plugins
Findings
After automated analysis a new finding will be added. Select this finding to view and edit its details
Threat Management
Identified Threats can be managed by manually assigning priorities and adding comments
Reporting
Reports can be exported as PDF, HTML or Markup files

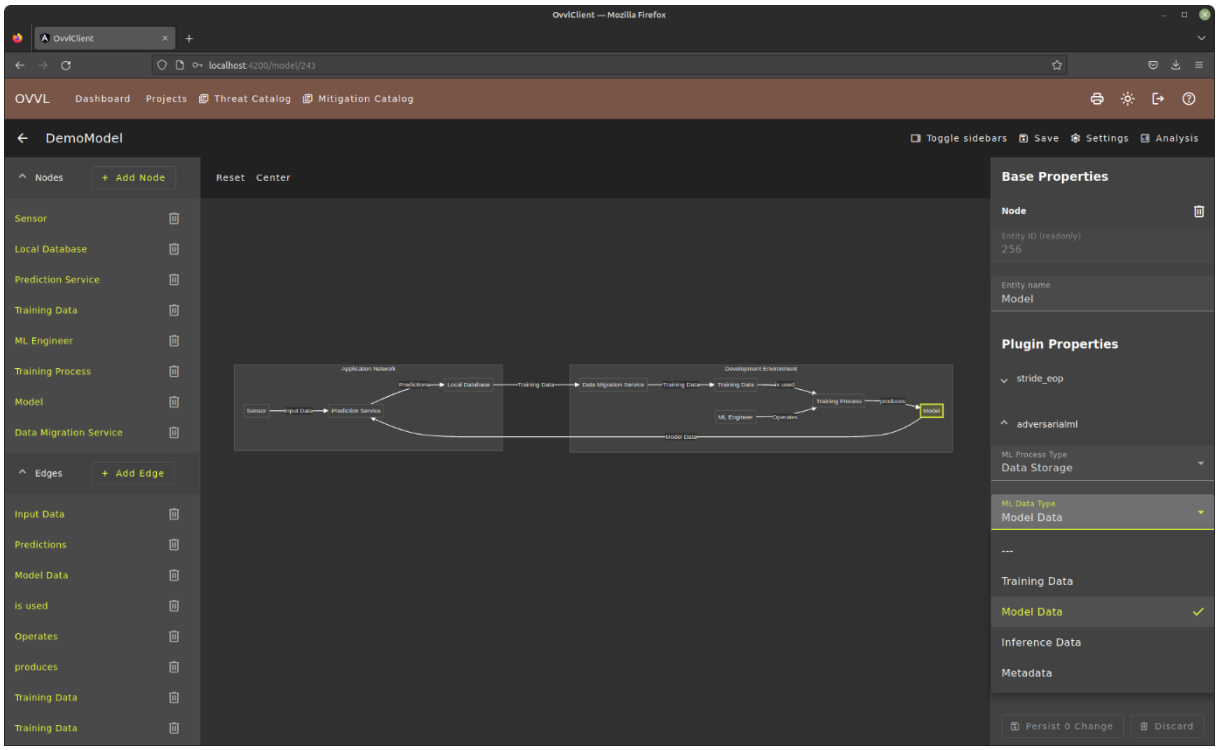

OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/model/243
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoModel
Toggle sidebars
Save
Settings
Analysis
Nodes
+ Add Node
Edges
+ Add Edge
Groups
+ Add Group
Reset
Center
Select an entity in the graph to show its properties
Model Settings
Plugins
Priorities
Priority Description
High
Priority Description
Medium
Priority Description
Low
New Priority Description
Save
Close


OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/model/243
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoModel
Toggle sidebars
Save
Settings
Analysis
Nodes
+ Add Node
Edges
+ Add Edge
Groups
+ Add Group
Reset
Center
Select an entity in the graph to show its properties
Model Settings
Plugins
STRIDE Extended
Active
Adversarial Machine Learning
Active
STRIDE
Active
LINDDUN
Active
Plugin Details
Id: adversarialml
Description:
Priorities
Save
Close

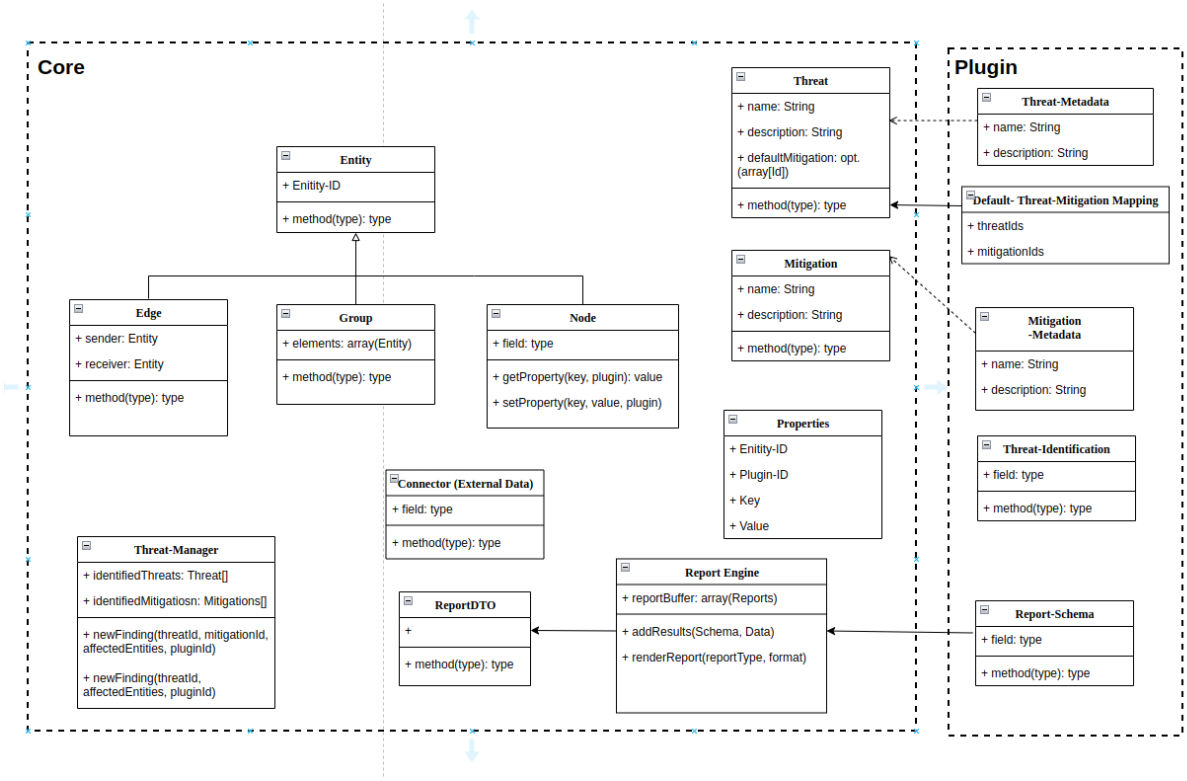

OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/project/242
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoProject
+ Create Model
Import Model
Name
Creator
Created At
Updated At
Interaction
DemoModel
TestUser1
2024-11-20 15:40:31
2024-11-20 15:40:31
Export
Rename
Delete

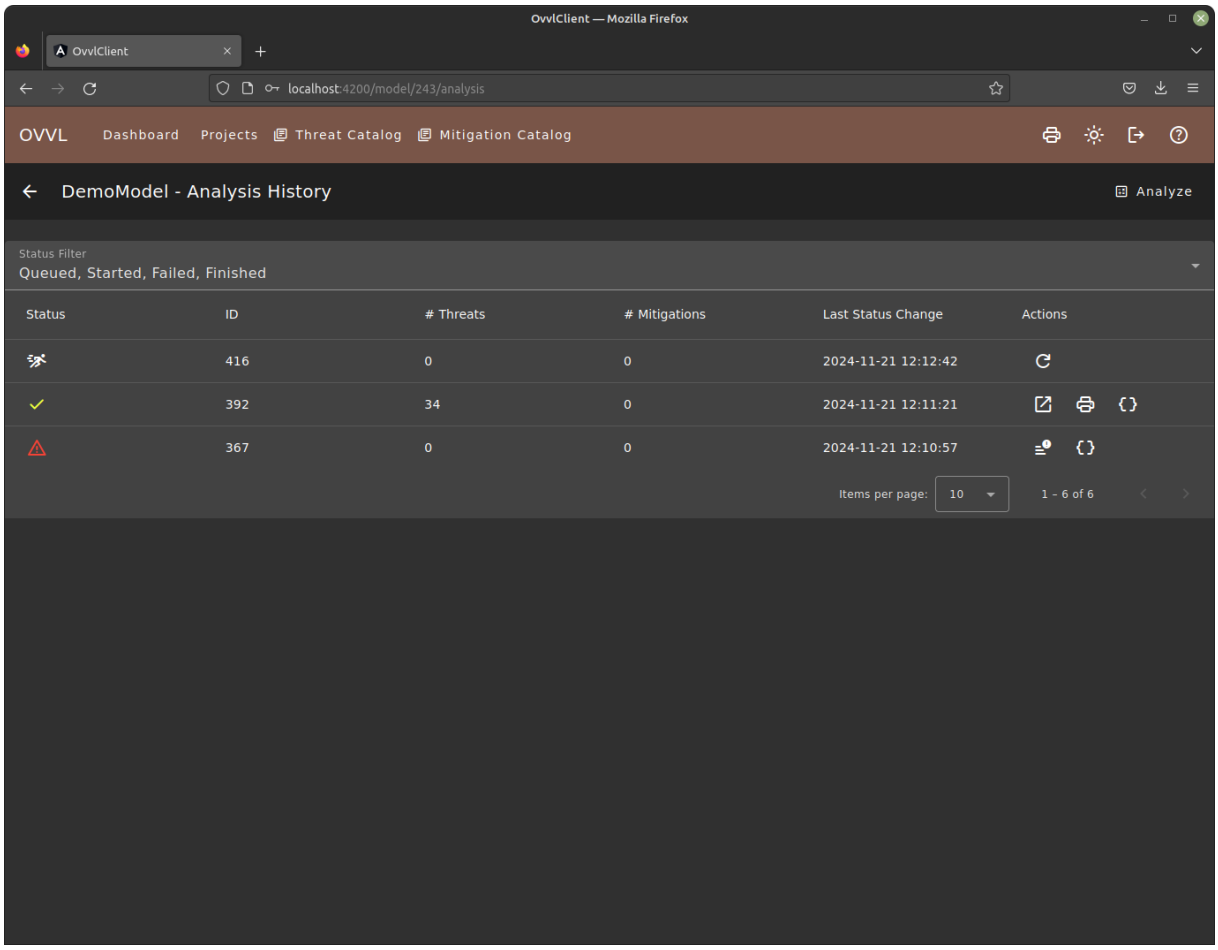

OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/model/243
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoModel
Toggle sidebars
Save
Settings
Analysis
Nodes
+ Add Node
Reset
Center
Sensor
Local Database
Prediction Service
Training Data
ML Engineer
Training Process
Model
Data Migration Service
Edges
+ Add Edge
Input Data
Predictions
Model Data
is used
Operates
produces
Training Data
Training Data
Base Properties
Node
Entity ID (readonly)
256
Entity name
Model
Plugin Properties
stride_eop
adversarialml
ML Process Type
Data Storage
ML Data Type
Model Data

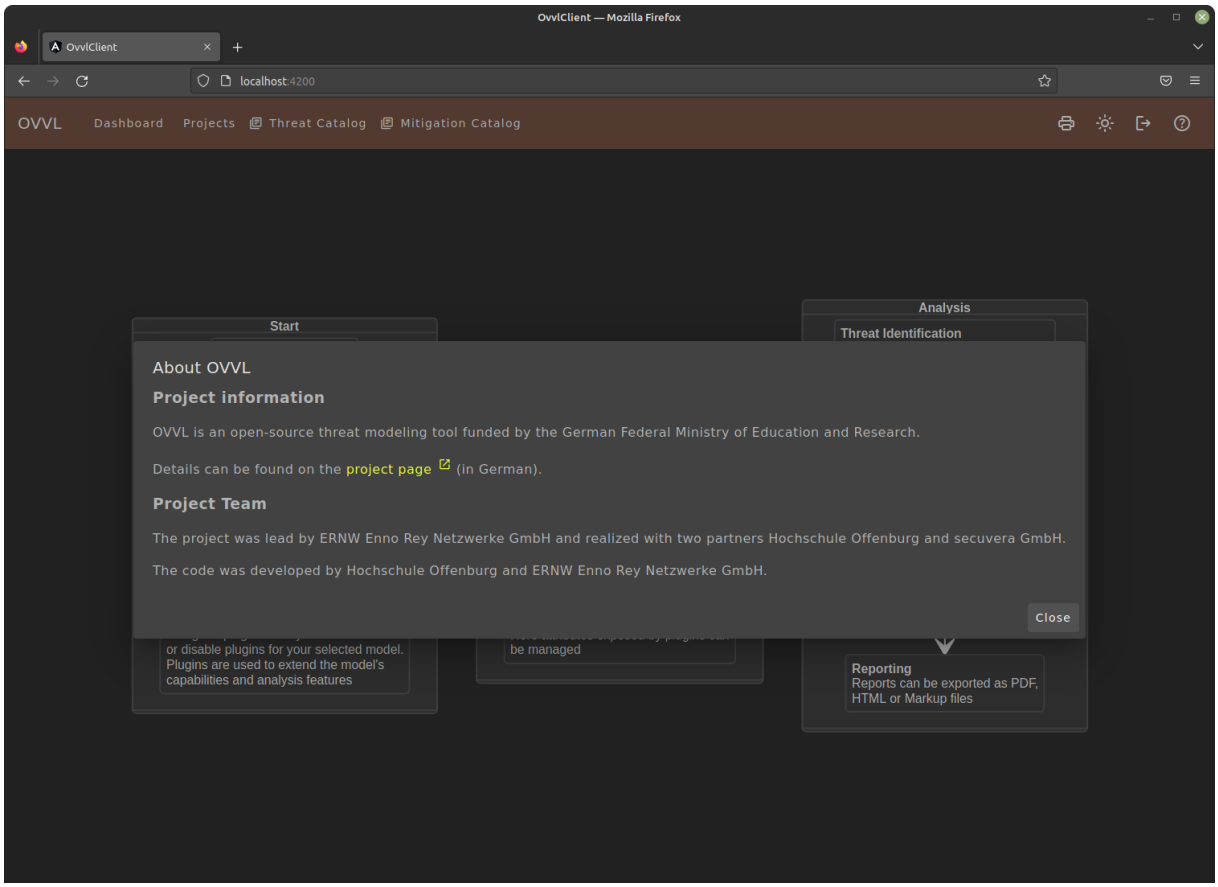
Training Data
Model Data
Inference Data
Metadata
Persist 0 Change
Discard

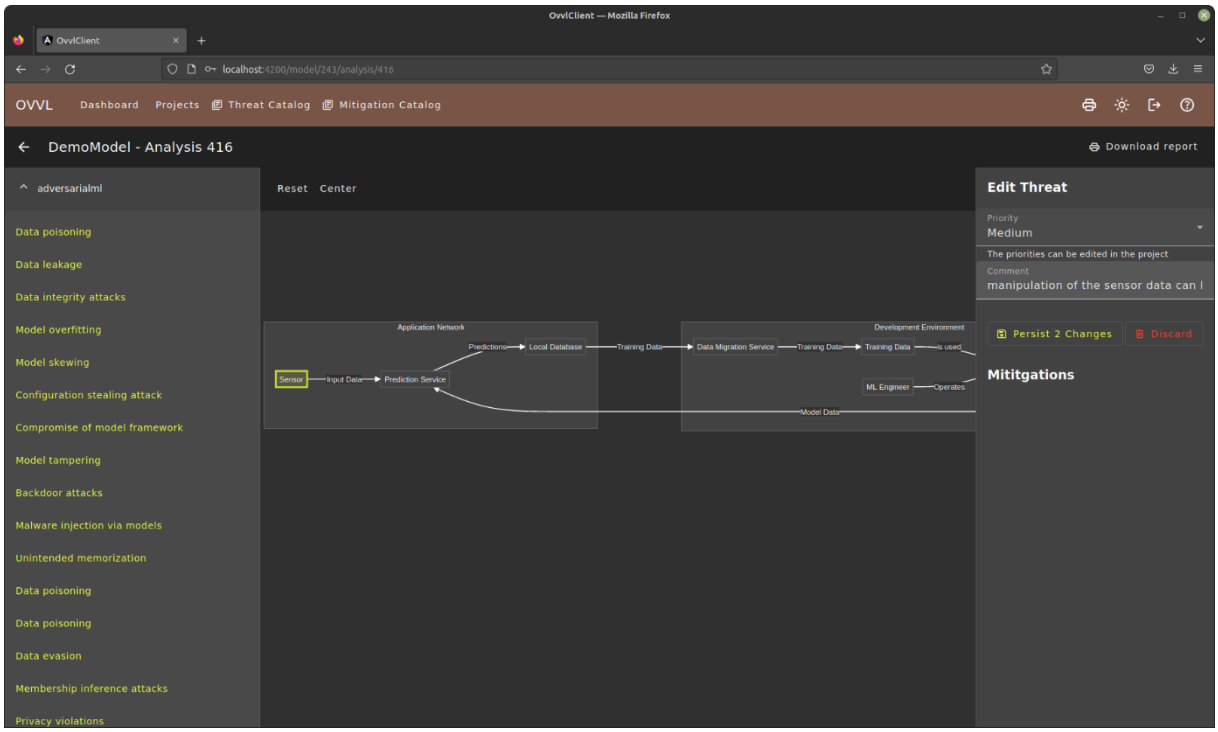

OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/model/243
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoModel
Toggle sidebars
Save
Settings
Analysis
Nodes
+ Add Node
Reset
Center
Sensor
Local Database
Prediction Service
Training Data
ML Engineer
Training Process
Model
Data Migration Service
Edges
+ Add Edge
Input Data
Predictions
Model Data
is used
Operates
produces
Training Data
Training Data
Application Network
Development Environment
Base Properties
Node
Entity ID (readonly)
256
Entity name
Model
Plugin Properties
stride_eop
adversarialml
ML Process Type
Data Storage
ML Data Type
Model Data

Training Data
Model Data
Inference Data
Metadata
Persist 0 Change
Discard


Core
Entity
+ Enitity-ID
+ method(type): type
Edge
+ sender: Entity
+ receiver: Entity
+ method(type): type
Group
+ elements: array(Entity)
+ method(type): type
Node
+ field: type
+ getProperty(key, plugin): value
+ setProperty(key, value, plugin)
Threat
+ name: String
+ description: String
+ defaultMitigation: opt. (array[Id])
+ method(type): type
Mitigation
+ name: String
+ description: String
+ method(type): type
Properties
+ Enitity-ID
+ Plugin-ID
+ Key
+ Value
Connector (External Data)
+ field: type
+ method(type): type
Threat-Manager
+ identifiedThreats: Threat[]
+ identifiedMitigatiosn: Mitigations[]
+ newFinding(threatId, mitigationId, affectedEntities, pluginId)
+ newFinding(threatId, affectedEntities, pluginId)
ReportDTO
+
+ method(type): type
Report Engine
+ reportBuffer: array(Reports)
+ addResults(Schema, Data)
+ renderReport(reportType, format)
Plugin
Threat-Metadata
+ name: String
+ description: String
Default- Threat-Mitigation Mapping
+ threatIds
+ mitigationIds
Mitigation -Metadata
+ name: String
+ description: String
Threat-Identification
+ field: type
+ method(type): type
Report-Schema
+ field: type
+ method(type): type


OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/model/243/analysis
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoModel - Analysis History
Analyze
Status Filter
Queued, Started, Failed, Finished
Status
ID
Threats
Mitigations
Last Status Change
Actions
416
0
0
2024-11-21 12:12:42
392
34
0
2024-11-21 12:11:21
367
0
0
2024-11-21 12:10:57
Items per page:
10
1 - 6 of 6


OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
Start
Analysis
Threat Identification
About OVVL
Project information
OVVL is an open-source threat modeling tool funded by the German Federal Ministry of Education and Research.
Details can be found on the project page (in German).
Project Team
The project was lead by ERNW Enno Rey Netzwerke GmbH and realized with two partners Hochschule Offenburg and secuvera GmbH.
The code was developed by Hochschule Offenburg and ERNW Enno Rey Netzwerke GmbH.
Close
or disable plugins for your selected model. Plugins are used to extend the model's capabilities and analysis features
be managed
Reporting
Reports can be exported as PDF, HTML or Markup files


OvvlClient — Mozilla Firefox
OvvlClient
localhost:4200/model/243/analysis/416
OVVL
Dashboard
Projects
Threat Catalog
Mitigation Catalog
DemoModel - Analysis 416
Download report
adversarialml
Reset
Center
Data poisoning
Data leakage
Data integrity attacks
Model overfitting
Model skewing
Configuration stealing attack
Compromise of model framework
Model tampering
Backdoor attacks
Malware injection via models
Unintended memorization
Data poisoning
Data poisoning
Data evasion
Membership inference attacks
Privacy violations
Application Network
Development Environment
Sensor
Input Data
Prediction Service
Predictions
Local Database
Training Data
Data Migration Service
Training Data
Training Data
is used
ML Engineer
Operates
Model Data
Edit Threat
Priority
Medium
The priorities can be edited in the project
Comment
manipulation of the sensor data can l
Persist 2 Changes
Discard
Mititgations

Forschungsrahmenprogramm der Bundesregierung für IT-Sicherheit – Selbstbestimmt und sicher in der digitalen Welt

Verbundprojekt:

Automatisierte Bedrohungsmodellierung für den sicheren Softwarelebenszyklus - OVVL

Teilvorhaben: ***OVVL - Komponentenbasiert, Offen und Modular - KOM***

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Abschlussbericht (Kurzfassung) Hochschule Offenburg

Förderkennzeichen: **16KIS1403**
Laufzeit des Vorhabens:
01.07.2021 bis 30.06.2024

Zuwendungsgeber:
Bundesministerium für
Bildung und Forschung

Zuwendungsempfänger:
Hochschule Offenburg

Hochschule Offenburg
offenburg.university

Prof. Dr. Andreas Schaad Offenburg, 19.12.2024

Das Projekt „OVVL“ wurde erfolgreich beendet. Ziel des Projekts OVVL war die automatisierte Unterstützung eines „Secure by Design“ Ansatzes, der den gesamten sicheren Softwarelebenszyklus adressiert. Dafür wurde folgende drei Teilprobleme gelöst:

1. Erstellung einer Methodik und unterstützender Tools für automatisierte (machine-learning basierte) Bedrohungsanalysen von Softwarearchitekturen in der Designphase.
2. Entwurf von Datenstrukturen und offenen Schnittstellen, um die Ergebnisse solch einer Analyse auch von automatisierten Verfahren in späteren Phasen des sichereren Softwarelebenszyklus nutzen zu können (insb. Testen und Vulnerability Management).
3. Entwurf eines Prozessportfolios, um Methodik, Tooling und Schnittstellen in domänenspezifische Bedrohungsanalysen (Automatisierungstechnik, Gesundheitswesen, IT, …) und den sicheren Entwurf genereller Software einbinden zu können.

Diese Arbeiten resultieren in dem OVVL Datenmodell und Tool, welches nun durch die beteiligten Partner ENRW und Secuvera validiert und ggf. verstetigt wird:

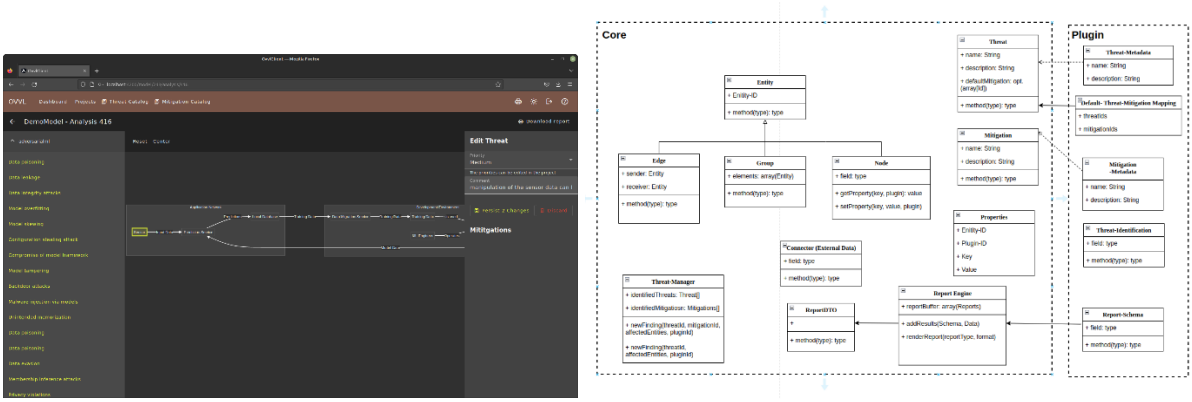


Abbildung: OVVL Datenmodell und Tool

Folgende Veröffentlichungen wurden zudem aus dem Projekt generiert:

1) Andreas Schaad, Dominik Binder: Deep-Learning-based Vulnerability Detection in Binary Executables. 15th International Symposium on Foundations & Practice of Security (FPS 2022), Ottawa, Kanada
2) Andreas Schaad, Dominik Binder: Deep-Learning-based Vulnerability Detection in Binary Executables. CoRR abs/2212.01254 (2022) – erweiterte Version

Ein in Teilen (KI-basierte Analyse) auf OVVL basierendes Folgeprojekt wurde durch die HSO erfolgreich initiiert (3 jähriges KMU Innovativ Projekt „SMILE“ FKZ 16KIS1929, 01.08.2023-31.07.2026).